(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 August 2002 (29.08.2002)

PCT

(10) International Publication Number
WO 2002/067173 A1

(51) International Patent Classification[7]: G06F 17/60, H04L 12/24

(21) International Application Number: PCT/SG2002/000027

(22) International Filing Date: 22 February 2002 (22.02.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PR 3331 23 February 2001 (23.02.2001) AU

(71) Applicant *(for all designated States except US)*: **I-SPRINT INNOVATIONS PTE LTD** [SG/SG]; 750C Chai Chee Industrial Park, #02-14/15 Chai Chee Road, Singapore 469003 (SG).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **GU, Guoliang** [CN/SG]; c/o i-Sprint Innovations Pte Ltd, 750C Chai Chee Industrial Park, #02-14/15 Chai Chee Road, Singapore 469003 (SG). **CHAN, Kim Hing** [SG/SG]; c/o i-Sprint Innovations Pte Ltd, 750C Chai Chee Industrial Park, #02-14/15 Chai Chee Road, Singapore 469003 (SG).

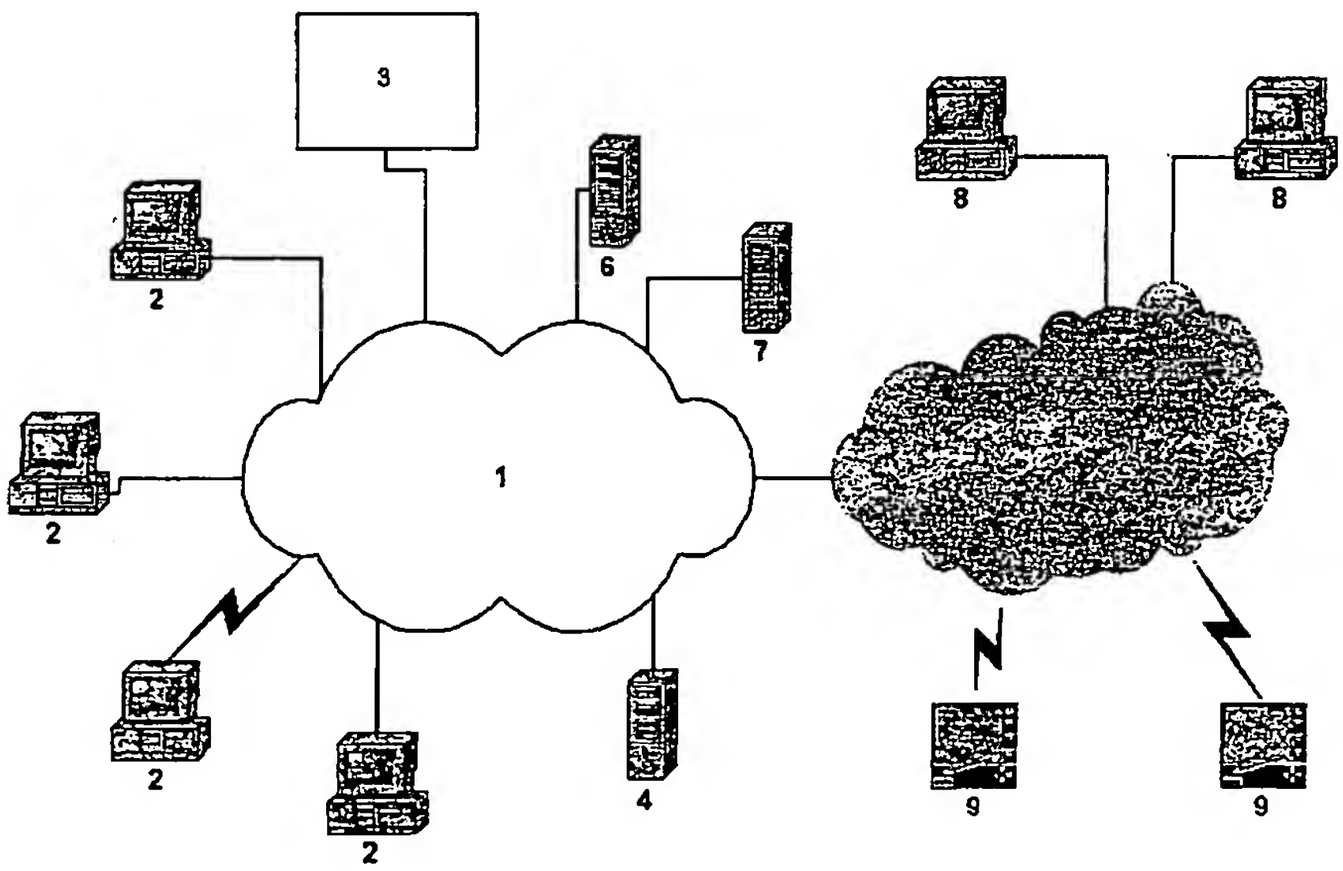
(74) Agent: **SIM, Yuan Meng, Andrew**; Shook Lin & Bok, 1 Robinson Road, #18-00, AIA Tower, Singapore 048542 (SG).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent

[Continued on next page]

(54) Title: A HIERARCHY MODEL



(57) Abstract: The present invention relates to a hierarchy model for modelling the security policy of a business. The model includes a number of segment nodes (61, 62, 63, 64, 65, 66, 67, 68) which define the security policy that applies to a respective segment of the business. Principal, group and application nodes (P1, P2, P3, P4, G1, G2, G3, G4, A1, A2, A3, A4), are then used to define the security policy which applies to respective individuals, group of individuals and applications or objects within the business. A number of connections are then used to connect the nodes so that the connected nodes reflect the structure of the business.

WO 2002/067173 A1

(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

(48) Date of publication of this corrected version:
27 October 2005

(15) Information about Correction:
see PCT Gazette No. 43/2005 of 27 October 2005, Section II

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

A HIERARCHY MODEL

Background of the Invention

The present invention relates to a hierarchy model for modelling the security policy of a business, and in particular, to a hierarchy model which when implemented is capable of allowing security administrators to centrally manage user and application security related attributes (authentication methods, access rights, privileges, access control rules) and other application attributes.

Description of the Prior Art

The reference to any prior art in this specification is not, and should not be taken as, an acknowledgment or any form of suggestion that the prior art forms any part of the common general knowledge in Australia

Security administration in a large-size, mission-critical and security-sensitive environment is a big challenge when many different applications (web servers, application servers, etc), and thousands of different users (employees, partners, customers) need be managed.

In particular, it is necessary to consider:

1) How to define and enforce an enterprise-wide security policy in a large organization?
2) How to support separation of duties to prevent giving too much power to one person or functional role?
3) How to manage thousands of different users (employees, partners, customers), their credentials and privileges? and,
4) How to manage different web-enabled mission-critical and security-sensitive applications?

There are very few research papers in the security administration area, although a number of attempts have been made by the industry to implement simple security administration models. These include using techniques such as:

1) Having individual departments manage their own business applications;
2) Using distributed security system with separate security domains for different departments or/and business applications;
3) Providing centralized security administration; and,
4) Delegating some or all the security administration.

However, these solutions suffer from a number of drawbacks.

In the first case, if individual departments are allowed to manage their security policies, then it is impossible to define and enforce enterprise wide security policies due to the logistics of the problem. This can have significant drawbacks if different departments end up with significantly different policies that are effectively incompatible, as this can lead to the security of one or more departments being compromised.

In the second example, reasonable security can be achieved for separate or less-connected businesses. However, cross-domain communications, particularly across different web applications, will remain a problem. In addition to this, it is again difficult to define and enforce enterprise wide security

policies.

In the third case, it is practically unfeasible to manage a large number of users and multiple applications in a distributed, heterogeneous computing environment. In addition to this, security administrators are required to know multiple inconsistent administration interfaces that eventually become bottlenecks to the implementation of a business wide policy.

Finally in the fourth case, granular administration rights are not defined. Furthermore, effective delegation scopes are not clear and accordingly, delegation without granularity will not allow for effective overall control.

Accordingly, enterprise-wide security policy cannot be easily and effectively enforced by current security systems. As a result, organizations resort to manual procedures for segregation of duties and security privilege enforcement, thus reducing the effectiveness of any security policy.

In addition to this, current systems are unable to securely track the day to day activities of users of the security policy. Accordingly, when a security problem arises, it can be difficult to determine where and when the breach of policy occurs.

As a result, there is need for a security administration solution to manage access control or authorisation decisions and security policies in multiple-tier, mission-critical application environments.

Summary of the Present Invention

In a first broad form, the present invention provides a hierarchy model for modeling the security policy of a business, the hierarchy model including:

a) A number of segment nodes, each segment node being used to define the security policy which applies to a respective segment of the business, each segment being associated with a number of respective individuals and/or applications, each segment node including at least one of the following:

i) A principal node, the principal nodes being used to define the security policy which applies to respective individuals within the business, each principal node including one or more principal entities, and each principal entity corresponding to a respective individual;

ii) An application node, the application nodes being used to define the security policy which applies to respective applications or objects within the business, each application node including one or more application or object entities, and each application or object entity corresponding to a respective application or object;

iii) A group node, the group nodes being used to define the security policy which applies to respective groups of principals within the business, each group node including one or more group entities, and each group entity corresponding to a group of respective individuals;

b) A number of connections, the connections defining associations between the nodes to reflect the structure of the business.

In a second broad form, the present invention provides a method of generating a hierarchy model for modeling the security policy of a business, the method including:

a) Defining the business as a number of segments, each segment being a respective unit within the business;
b) Determining the individuals, groups of individuals and/or applications associated with each unit;
c) Defining a number of segment nodes, each segment node being used to define the security policy which applies to a respective segment of the business;
d) Defining at least one principal, group and/or application node, each principal, group and/or application node being used to define the security policy which applies to respective individuals, groups of individuals and/or applications; and,
e) Defining a number of connections between the defined nodes in accordance with the relationships between the segments, individuals and/or applications to reflect the structure of the business.

In a third broad form, the present invention provides a computer program product including computer executable program code for generating a hierarchy model in accordance with the method of the first broad form of the invention.

In a fourth broad form the present invention provides a system for generating a hierarchy model for modeling the security policy of a business, the hierarchy model including:

a) An input, adapted to receive inputs:
   i) Defining the business as a number of segments, each segment being a respective unit within the business;
   ii) Identifying the individuals and/or applications associated with each unit;
   iii) Defining a number of segment nodes, each segment node being used to define the security policy which applies to a respective segment of the business;
   iv) Defining at least one principal, group and/or application node, each principal, group and/or application node being used to define the security policy which applies to respective individuals, groups of individuals and/or applications; and,
   v) Defining a number of connections between the defined nodes in accordance with the relationships between the segments, individuals and/or applications to reflect the structure of the business.
b) A store; and,
c) A processor adapted to:
   i) Generate a number of tables in the store;
   ii) Associate each table with a respective node in the hierarchy model; and,
   iii) Define a number of links between the tables, each link representing a connection between respective nodes within the tables.

Accordingly, the present invention provides a hierarchy model for modelling security policy of a business, as well as a method, a computer program product and a system for generating such a model.

The model includes a number of segment nodes, which can in turn include a number of principal, application or group nodes. The business is modeled in terms of a number of interconnected segment nodes, with the principal, group, and application nodes being used to represent individuals, groups of individuals, and applications/objects within the business respectively. This in turn allows the security

policy to be defined for the segments of the business separately or in conjunction with other segments, allowing a security policy to be defined for smaller units of a business separately.

This technique can be extended so that the segment include related external organisations, such as external suppliers, thereby allowing the hierarchy model to be used in modelling the security policy of a business on its own, or of a business and its related external organisations.

Using this technique the defined policy can apply to the business as a whole, or to different units, such as departments within the business, as well as related external organisations. Respective security administrators can then be used to define variations in the overall policy to each segment. Policies applied to a given segment are then applied to the associated individuals, groups or applications principal so that the security policy applies to all entities within the entire business.

The respective individual may correspond to a functional (non-interactive) identity within the business. Typically, within a multi tier environment there are one or more servers, daemons or the like, which need to login to the security system in order to perform actions, or complete tasks, on behalf of users of the system (individuals). Accordingly, in order to be able to login to the security system, the servers, daemons or the like, need respective identities. As the servers or the like are unable to handle interactive login prompts, these identities are referred to as non-interactive ids. Accordingly, the individuals may comprise a server, daemon or the like.

Thus it will be appreciated that the hierarchy model can be used for modelling security administration, as well as security policy.

Typically each segment node may include one or more child segment nodes, each child segment node defining the security policy for a logical unit within the respective parent segment. Thus, for example, a parent segment could be a regional office with the child segment forming a department within the office.

The hierarchy model usually further includes at least one root segment node, the root segment node being a segment node with no parent nodes. This can be used to represent the security policy for the overall business, with variations from this policy being defined in the child segments as required. Thus, this allows common security policies which would be applied to all segments to be factored out into a single root segment node.

The model typically further includes application collection entities, each application collection entity representing a collection of application entities. Again, this allows common security policies which would be applied to a number of applications to be factored out into application collections..

Usually the model is implemented within a database, the database including a number of tables, each table representing a respective type of node within the model. This can allow the database structure to reflect the structure of the hierarchy model allowing changes to the hierarchy model to be easily implemented within the database. Furthermore, transfer of security policy from one level of the hierarchy model to another can easily be implemented by propagating data between the tables in the database. This allows the present invention to be implemented using a relational database, or the like.

Accordingly, the database usually includes a segment table, with each segment node being represented by a respective entry within the segment table.

Similarly a principal table is usually provided with each principal entity being represented by a respective entry within the principal table, a group table, with each group entity being represented by a respective entry within the group table and an applications table, with each application entity, object entity and application collection entity being represented by a respective entry in the application table.

Furthermore, the number of connections are preferably represented by links between entries in the tables.

Preferably each node within the model has an associated number of security settings, the security policy being defined by selecting appropriate ones of the security settings. In this case, each type of node can have respective security settings. Typically, a number of standard security settings will be pre-defined associated with each model. However, it will also typically be possible to add in additional security settings should this be necessary.

Typically the security settings or other application attributes available including settings relating to password length, expiry duration, minimum age, and the like, although restrictions on access to applications and information can also be defined.

Typically each entry is defined in a respective row of the respective table, with each security setting being defined within a respective column.

Typically each node is defined in a respective row of the respective table, the security setting being defined in a respective column. Thus, the database will typically contain a segment table that includes details of each segment in a respective row of the table. The security settings for each segment can then be defined within respective columns such that the security settings for all the nodes at a given level within the hierarchy are defined in a single table. However, any database structure, could be used.

In a fifth broad form, the present invention provides a method of defining a security policy for a business, the method including:

a) Defining the business in terms of a hierarchy model, the hierarchy model including:

   i) A number of segment nodes, each segment node being used to define the security policy which applies to a respective segment of the business;

   ii) At least one principal, group and/or application node, each principal, group and/or application node being used to define the security policy which applies to respective individuals, groups of individuals and/or applications; and,

   iii) A number of connections between the defined nodes, the connections representing the relationships between the segments, individuals and/or applications to reflect the structure of the business.

b) Specifying the security policy for at least some of the nodes in the hierarchy model; and,

c) Propagating the specified security policy to other nodes in the hierarchy model.

In a sixth broad form, the present invention provides A computer program product including computer executable code for defining a security policy for a business in accordance with the method of the fifth broad form of the invention.

In a seventh broad form, the present invention provides a system for defining a security policy for a business or extend organisations, the system including:

a) A system for defining the business in terms of a hierarchy model, the hierarchy model including:
   i) A number of segment nodes, each segment node being used to define the security policy which applies to a respective segment of the business or extended organisations;
   ii) At least one principal, group and/or application node, each principal, group and/or application node being used to define the security policy which applies to respective individuals, groups of individuals and/or applications; and,
   iii) A number of connections between the defined nodes, the connections representing the relationships between the segments, individuals and/or applications to reflect the structure of the business.
b) An input, adapted to receive inputs specifying the security policy for at least one of the nodes in the hierarchy model; and,
c) A store for storing the specified security policy; and,
d) A processor adapted to propagate the specified security policy to other nodes in the hierarchy model.

Accordingly, the present invention also provides a method, a computer program product and a system for defining a security policy for a business. This can also be extended for use with extended organisations.

This generally uses the hierarchy model of the first broad form of the present invention. Accordingly, the technique of the invention involves defining the businesses in terms of the hierarchy model, specifying the security policy for some nodes and, propagating the specified security policy to other nodes.

Typically the method of specifying the security policy for a node includes:

a) Generating security data defining security settings for the respective node; and,
b) Inputting the security data as part of the respective entry within the respective table within the database.

The method usually further involves propagating the defined security settings to other nodes in the hierarchy model by propagating the security data to other linked entries in the database. Thus, this provides a technique for having the database automatically updated by a propagating security data throughout the database. This propagation is carried out in such a way that the security policy can be defined for segments and then propagated to child segments automatically. However, the security settings applied to a child segment node may override the security settings propagated from the respective parent segment node. In this circumstance, the method can be controlled to prevent the security policy applied to a child segment from being weaker than the security policy of the parent segment.

In a eighth broad form, the present invention provides a method of implementing a security policy for a business, the method including:

a) Defining a security policy for the business, the security policy being defined in terms of a hierarchy model including:
   i) A number of segment nodes, each segment node representing a respective segment of the business;

ii) At least one principal, group and/or application node, each principal, group and/or application node representing a respective individual, group of individuals and/or applications within the business; and,

iii) A number of connections, the connections defining the associations between the nodes, the security policy being defined by security settings associated with each node;

b) Monitoring action to be taken by a respective segment, individual or application within the business;

c) Determining if the action is permissible in accordance with the defined security settings; and,

d) Allowing or preventing the action in accordance with the result of the determination.

In a ninth broad form, the present invention provides a computer program product including computer executable code for implementing a security policy in accordance with the method of the eighth broad form of the present invention.

In a tenth broad form, the present invention provides a system for implementing a security policy for a business, the system including:

a) A system for defining a security policy for the business, the security policy being defined in terms of a hierarchy model including:

i) A number of segment nodes, each segment node representing a respective segment of the business;

ii) At least one principal, group and/or application node, each principal, group and/or application node representing a respective individual, group of individuals and/or applications within the business; and,

iii) A number of connections, the connections defining the associations between the nodes, the security policy being defined by security settings associated with each node; and,

b) A processor, the processor being adapted to:

i) Monitor action to be taken by a respective segment, individual or application within the business;

ii) Determine if the action is permissible in accordance with the defined security settings; and,

iii) Allow or prevent the action in accordance with the result of the determination.

Accordingly, the present invention also provides a method, a computer program product and a system for implementing a security policy. This is achieved by monitoring actions to be taken by respective segments, individuals or applications within the business and then determining if this action is permissible in accordance with security settings defined within the hierarchy model. This then allows the action to be prevented or allowed depending on the security settings defined.

Typically the security settings are stored in a database associated with the respective segment, principal, group and/or object node, the method of determining if the action is permissible including:

a) Accessing the database to determine the security settings for the respective segment, individual and/or application; and,

b) Comparing the accessed security settings to the action to be taken to determine if the action is permitted.

Preferably, the hierarchy model is a model according to the first broad form of the present invention.

In an eleventh broad form the present invention provides a method of controlling the security policy of a business, the security policy being defined using a hierarchy model including:

a) A number of segment nodes, each segment node having an associated number of security settings, the segment node being used to define the security policy which applies to a respective segment of the business by selecting appropriate ones of the security settings; and,

b) A number of connections, the connections defining associations between the segment nodes to reflect the structure of the business, the method including:

i) Assigning an administrator to each segment; ,

ii) Defining administration rights that can be assigned to any administrator, the administration rights controlling the administration procedures the respective administrator can perform; and,

iii) Assigning appropriate ones of the administration rights to the administrator, if required.

In a twelfth broad form the present invention provides a computer program product for controlling the administration of the security policy of a business, the computer program product including computer executable code for performing the method of the eleventh broad form of the present invention.

In an thirteenth broad form the present invention provides a system for controlling the administration of the security policy of a business, the security policy being defined using a hierarchy model including:

a) A number of segment nodes, each segment node having an associated number of security settings, the segment node being used to define the security policy which applies to a respective segment of the business by selecting appropriate ones of the security settings; and,

b) A number of connections, the connections defining associations between the segment nodes to reflect the structure of the business, the system including:

i) An input for:

(1) Receiving an indication of an administrator assigned to each segment; ,

(2) Defining administration rights that can be assigned to any administrator; and,

(3) Assigning appropriate ones of the administration rights to the administrator, if required; and,

ii) A processor adapted to control the administration procedures the respective administrator can perform in accordance with the assigned administration rights.

Accordingly the present invention seeks to provide a method, computer program and system for controlling the administration of the security policy of a business. This is achieved by having administrators assigned to respective segments, and then controlling the action that can be taken by the administrators. This is achieved by the use of granular administration rights which control the administration actions an administrator can perform, including the rights to alter security policy and settings, as well as the rights to read/view various policy settings.

Typically the method of assigning the administrator includes defining an administrator entity for each segment, the administrator entity representing a respective individual within the business.

In this case, the method of controlling the administration rights usually includes assigning appropriate ones of the administration rights to the defined administrator entity.

Preferably the administration rights are defined at the system level.

In a fourteenth broad form the present invention provides a method of controlling the administration of the security policy of a business, the security policy being defined using security settings, the method involving:

a) Assigning first and second administrators to control alteration of the security settings;

b) Controlling the first administrator to only allow the first administrator to propose changes to the security settings; and,

c) Controlling the second administrator to only allow the second administrator to accept or reject proposed changes to the security settings, thereby allowing the first and second administrators to control the security settings.

In a fifteenth broad form the present invention provides a computer program product including computer executable code for controlling the administration of the security policy of a business in accordance with the method of the fourteenth broad form of the present invention.

In a sixteenth broad form the present invention provides a system for controlling the administration of the security policy of a business, the security policy being defined using security settings, the system including:

a) An input for assigning first and second administrators to control alteration of the security settings; and

b) A processor adapted to:

i) Control the first administrator to only allow the first administrator to propose changes to the security settings; and,

ii) Control the second administrator to only allow the second administrator to accept or reject proposed changes to the security settings, thereby allowing the first and second administrators to control the security settings.

Typically the method of assigning the first and second administrators includes defining first and second administrator entities, the first and second administrator entities representing respective individuals within the business.

The method of controlling the administrators usually involves:

i) Defining a number of administration rights that can be assigned to any administrator entity, the administration rights controlling the administration procedures the respective administrator can perform; ; and,

ii) Assigning appropriate ones of the administration rights to the respective administrator entity.

The administration rights are usually selected ones of the security settings and wherein the administration rights are selected to prevent an administrator altering the administration rights of their own respective administrator entity.

Typically the hierarchy model is a model according to the first broad form of the present invention.

In a seventeenth broad form the present invention provides a method of controlling the administration of the security policy of a business, the security policy being defined using a hierarchy model including:

a) A number of segment nodes, each segment node having an associated number of security settings, the segment node being used to define the security policy which applies to a respective segment of the business by selecting appropriate ones of the security settings; and,

b) A number of connections, the connections defining associations between the segment nodes to reflect the structure of the business, the method including:

i) Assigning an administrator to each segment, the administrator being responsible for the security settings of the respective segment; and,

ii) Allowing a first administrator to delegate responsibility for the security settings of a respective segment to a second administrator.

In an eighteenth broad form the present invention provides a computer program product including computer executable code for controlling the administration of the security policy of a business in accordance with the method of the seventeenth broad form of the present invention.

In a nineteenth form the present invention provides a system for controlling the administration of the security policy of a business, the security policy being defined using a hierarchy model including:

a) A number of segment nodes, each segment node having an associated number of security settings, the segment node being used to define the security policy which applies to a respective segment of the business by selecting appropriate ones of the security settings; and,

b) A number of connections, the connections defining associations between the segment nodes to reflect the structure of the business, the system including:

i) An input for assigning an administrator to each segment, the administrator being responsible for the security settings of the respective segment; and,

ii) A processor adapted to allow a first administrator to delegate responsibility for the security settings of a respective segment to a second administrator.

Typically the method of delegating responsibility includes:

a) Determining if the administration rights for the first administrator allow responsibility to be delegated to the second administrator; and,

b) Selecting some or all of the administration rights for the second administrator to correspond to the administration right for the first administrator, thereby allowing the second administrator to perform some or all function of the first.

In a twentieth broad form the present invention provides a method of monitoring a security system implementing a security server, the method including:

a) Causing the security server to generate an indication of each action taken on the security system;

b) Digitally signing the indication of each action; and,

c) Storing the digitally signed indication.

In a twenty first broad form the present invention provides a computer program product including computer executable code for monitoring a security system in accordance with the method of the twentieth broad form of the present invention.

In a twenty second broad form the present invention provides a system for monitoring a security system implementing a security server, the system comprising:

a) An input for receiving an indication of each action taken on the security system;

b) A processor for digitally signing the indication; and,

c) A store for storing the digitally signed indication.

Accordingly, the present invention provides a method, computer program and a system for monitoring a security system. This is achieved by storing an indication, such as an audit log, recording actions that have be taken by the security system, thereby allowing the actions taken to be determined at a later date. The method usually includes storing the digitally signed indication in a table, each digitally signed indication being stored in a respective portion of the table.

Typically the security server is adapted to generate an authorisation or rejection indication in response to a request to perform an action, and wherein the method further includes causing the security server to generate an indication of each authorisation or rejection.

Digital signature ensures the integrity of the stored information. Preferably the store is located remotely to the security system on a dedicated system, such as one or more dedicated servers. This ensures the security of the stored information.

Accordingly, the security system based on above model offers a facility to centralize authentication, authorization and administration functions. The security system allows security administrators to centrally manage user and application security-related attributes, authentication methods (such as passwords, dynamic passwords or X.509 certificates), rights, privileges and access controls rules. This is achieved by using a central security system whenever applications need to access a resource on behalf of a user. The security server responds to the users or applications by referring to the policies and access control rules in its authorization database.

Brief Description of the Drawings

An example of the present invention will now be described with reference to the accompanying drawings, in which:-

Figure 1 is a schematic diagram of a network root segment including a security server according to the present invention;

Figure 2 is a schematic diagram showing the functionality of the security system of Figure 1;

Figure 3 is a schematic diagram of one of the security servers shown in Figure 2

Figure 4 is a schematic diagram of a hierarchy model according to the present invention;

Figure 5 is a screen shot taken from the graphical user interface (GUI) of the policy server of Figure 3, when defining the security policy of a segment node;

Figure 6 is a screen shot taken from the graphical user interface (GUI) of the policy server of Figure 3, when defining the privileges of a principal; and,

Figure 7 is a screen shot taken from the graphical user interface (GUI) of the policy server of Figure 3, when defining the access rights to an application.

Detailed Description of the Preferred Embodiments

An example of a system suitable for implementing the present invention is shown in Figure 1. As

shown the system includes a communications network 1 which forms the internal network of a business. Accordingly, the communications network 1 could range from a small Ethernet LAN (local area network), to a global WAN (wide area network) which is formed from a number of networks distributed throughout the world.

Coupled to the network 1 is a number of user end stations 2, which allow users to access the network and any of the services or information contained thereon. The end stations 2 may be any form of suitable processing system, and may be connected to the network 1 using either wired or wireless connections, as will be appreciated by persons skilled in the art.

In order to provide the log on security, a security system 3 is provided, as will be described in more detail below.

Also coupled to the network 1 is a web server 4 which is used to provide facilities such as a web site on the Internet 5, which is coupled to the network 1, as shown. In general, access to the Internet 5 is controlled using an appropriate firewall.

In addition to the user end stations 2 coupled to the network 1, access to the network can also be achieved in a number of different ways. Thus for example, the network 1 could be accessed using end stations 8 which are coupled to the Internet 5, via the Internet 5 and the web server 4. Alternatively, access could be achieved from mobile communications systems 9, such as a WAP or GPRS enabled mobile phone, or a lap top computer with a modem. In this case, access could be achieved by dialing into the network directly, or by establishing a connection via the Internet 5. In either of these cases, this will allow remote users, such as clients of the business, or employees located out of the office, to access the network 1.

The remainder of the description will be limited to describing the example of the end stations 2, although it will be appreciated that access using the end stations 8, or the mobile communications system 9 will be in a similar manner.

An application server 6 is provided to allow users of the end station 2 to access various applications, and a database server 7 is provided to allow the users of the end station 2 to access databases connected to the network 1.

The functionality of the security system 3 is shown in more detail in Figure 3. As shown, the security server includes a policy server 20, which is used to define a security policy for the business. An access server 23 is provided for implementing the defined security policy, whilst a service server 26 is used to configure and monitor the status of the other servers 20, 23.

The configuration of each of the servers 20, 23, 26 is shown in more detail in Figure 2. As shown, the servers typically include a network interface card 10 for coupling the server to the network 1. The network interface card 10 is in turn coupled to a processor 12 and a memory 13 via a bus 11. An input/output device (I/O device) 14 is also typically provided, usually in the form of a keyboard and monitor to allow information to be entered and obtained from the security server 3.

Accordingly, it will be appreciated that each of the servers 20, 23, 26 may be any form of processing system, such as a personal computer, or the like, which is configured in accordance with the present invention. This is typically achieved by having the processor 12 execute applications software

generated in accordance with the present invention which results in the functionality of the servers 20, 23, 26 being as shown in Figure 2. In this example, whilst three separate servers 20, 23, 26 are shown, it will be realized that the invention may be implemented on a single server, or on a larger number of servers distributed globally, depending on the circumstances.

Thus, as shown, the policy server 20 includes a policy daemon 21 and a report daemon 22, the access server 23 includes an authentication daemon 24, an authorization daemon 25, and an audit daemon 29. The service server 26 implements a configure/monitor 27 which operates to monitor and configure the other servers 20, 23, as required.

The daemons 21, 22, 24, 25, 29 are implemented within the processors 12 of the respective servers 20, 23, 26. In addition to this, each server 20, 23, 26 implements an API layer 30, which is a library of code for implementing a number of APIs. The API library code will form part of the applications software executed by the processors 12 of each of the servers 20, 23, 26, such that each of the servers 20, 23, 26 has access to a respective API layer 30, although for simplicity only a single API layer is shown in Figure 2.

In this example, the API layer includes a crypto-API 31 that is coupled to a key database 32 and a crypto library 33. The key database 32 and the crypto library 33 store the keys and rules which allow the each of the servers 20, 23, 26 to implement specific forms of cryptography.

A registry API 34 is provided to couple the servers 20, 23, 26 to a user database 35, and a policy database 36 as shown. The user database 35 is used to store information regarding each of the potential users of the system, such as user names (or IDs) and passwords, as will be explained in more detail below. Similarly, the policy database 36 is used to store the security policy of the business which the servers 20, 23, 26 define and then implement.

A log API 37 is provided to couple the servers 20, 23, 26 to a log store 38, the log store being used to maintain logs recording user access to the system, and a config API is provided to couple the configure/monitor 27 to a store of config files 40 which are used to configure the security server 3 for the particular operating environment

Accordingly, the API layer is used as an interface between the servers 20, 23, 26 and the key database 32, the crypto library 33, the user database 35, the policy database 36, the log store 38, and the config file store 40

Operation of the security system 3 according to the present invention will now be described.

In general, as outlined above, the security system 3 operates to enforce a security policy which is defined for an entire business.

In order to implement such a security policy, the security system 3 uses the access server 23 to be able to determine for each possible user of the end stations 2 the various access rights the respective user has. Thus, for example, certain users may only be provided with rights to access certain enterprise resources provided by the network system. Thus for example, use of the applications contained on the applications server 6, or use of some of the information available on the database server 7, may be limited for some users.

Accordingly, it is necessary for the access server 23 to be able to identify each user individually, as well as to be able to determine the access privileges that each user has.

The exact manner in which a user is identified may vary. Accordingly, the user may be asked to provide a user name and associated secret password, a user certificate, a dynamical password, or other suitable form of user identifier.

In general, the identification process is performed when the user attempts to access, for example, the web server 4 or the applications server 6. At this time, a Web Security Agent (WSA) contained in the web server 4, or an Application Security Agent (ASA) contained in the applications server 6, will prompt the user to enter their user identifier. The user identifier is entered by the user at the end station 2 before it is transferred to the WSA or ASA via the network 1.

Upon receipt of the user identifier, the WSA or ASA will forward the user identifier to the access server 23, which will operate to validate the request.

The access server 23 of the security server will then access the user database 35 to determine if the entered user identifier is valid. Thus, for example, if the user identifier is a user name and associated secret password, the access server will check that the entered password corresponds to the entered user name, allowing the identity of the user to be confirmed.

Once the identity of the user has been confirmed, the user is logged on to the system. At this stage, the access server 23 will assign a random session ID to the log in session of the user. Once this has occurred, the access server 23 will use the session ID to determine the current log in status of the user, as well as to determine the access privileges which are defined for the user in the security policy stored in the policy database.

Once this has been achieved, each time an action is to be taken, the WSA or ASA will query the access server 23, which will in turn access the authorization daemon to determine if the action is allowable within the scope of the users access privileges. An indication of this will be transferred to the WSA or the ASA, thereby allowing the WSA and the ASA to control the user's access to applications or information stored in the servers 4, 6, 7.

It will be appreciated from this that it is therefore necessary to define for each user of the end stations 2 the respective access privileges. This can be achieved by using the policy server 20 of the present invention to define a global security policy for the entire business organization.

In order to achieve this, the policy server 20 of the present invention allows a system administrator, or the like, to define a policy for the entire business organization by considering the business as a hierarchy structure formed from a number of different nodes.

The hierarchy structure generally includes segment nodes, principal nodes, group nodes, and applications nodes. The segment nodes are used by the policy officer to define security policies for different segments within the business. In general, a segment is taken to be a logical unit within the business, which is responsible for, or at least capable of, organising and administrating respective users, groups or applications.

Each segment node may contain zero or one principal nodes, zero or one group nodes, zero or one application nodes, and zero or more segment nodes.

If a first segment node contains a second segment node, the second segment node is sometimes referred to as a sub-segment, in that it relates to a smaller logical unit within the first segment. In this case,

the first segment and the second segment are related to each other via a parent child relationship, with the first segment node being the parent and the second segment node being the child.

The principal node is used to define access privileges for one or more individuals within the respective segment of the business. In the case, the individuals are referred to as principal entities and each principal node may contain zero or more principal entities.

The group nodes are provided to set access privileges for groups of individuals. In general, each group node may or contain zero or more group entities, with each group entity corresponding to a respective group of individuals.

The application node is used to define access permissions for application entities and may contain zero or more application entities or application collection entities. Each application collection entity may contain one or more application entities, whilst each application entity may contain one or more object entities. In this regard, the object entities may be any form of data object within the network system, and accordingly, the application entities (which are typically formed from one or more object entities) may correspond to software applications utilised by the system shown in Fig. 1.

In use, the policy officer therefore defines the security policy for a business by defining security policies at different levels within the hierarchy structure.

Accordingly, in order to define a security policy for the business under consideration, the policy officer must first define the business in terms of a hierarchy structure which includes a number of segment, principal, group and application nodes.

An example of such a structure is shown in Fig. 4.

As shown the hierarchy structure includes a number of segment nodes, 60, 61, 62, 63, 64, 65, 66, 67, 68. In this example, the segment node 60 has four child segment nodes (or sub-segments) 61, 62, 63, 64 which are connected to the segment node 60 via respective relationships 61a, 62a, 63a, 64a. In this example, the segment node 60 does not include a parent segment node and accordingly, the segment node 60 is referred to as a root segment node of the business, which represents the highest level at which security policy can be defined for this area of the business.

As shown, the child segment 61 is in itself a parent segment of the segment node 65, which is connected to the segment node 61 via the relationship 65a. Similarly, the segment node 64 is also the parent segment for the segment nodes 66, 67, 68, as shown by the relationship 66a, 67a, 68a.

As mentioned above, each segment can include zero or one principal nodes which allow the access privileges of individuals to be set. In this example, the segments 62, 65 are each shown as having a respective principle node 71, 72. Typically other segments will include a respective principle node, although this has not been shown for clarity purposes.

The individuals, whose access privileges are defined by the principal nodes, are shown for example, at P1, P2, P3, P4. Thus, as shown it is possible for each principal node to specify the access privileges for a number of principal entities.

A number of group nodes 73, 74, 75 are also shown. Again, each segment will only have zero or one group nodes. The group nodes are used to set access privileges for one or more group entities G1, G2, G3, G4 which in turn specify the access privileges of one or more principal entities. Thus, a principal

entity P1 may join one or more group entities G1, G2, G4 as appropriate.

Thus, for example, if the principal entity "Bob" represents an individual who is both an employee of the company and a manager, then the "Bob" principal entity may belong to both a "Staff" group entity, which specifies access privileges for general staff, and a "Manager" entity which specifies access privileges for managers.

Similarly, a number of application nodes 81, 82, 83, 84 are shown, each of which includes a number of application entities A1, A2, A3, A4. The application nodes are used to define the access permissions for the application entities, which as described above can correspond to applications software, or to data objects on the system, as shown by the object entities O1, O2 which make up the application entity A4.

Accordingly, the policy officer uses the structure of the segment, group, principal and application nodes to model the business structure.

As mentioned above, the segment nodes 61, 62, 63, 64 are used to represent different portions of the business. Typically, each segment node is an independent unit, which is capable of administering its own applications and users, and which accordingly can function autonomously independent of the other segment nodes. Accordingly, the administration units typically correspond to different departments, or the like, within the business. It is also possible that some or all of the administration unit within the hierarchy are not within the organization but those of the external organizations e.g. suppliers, customers and other business partners.

In this case, the segment nodes 65, 66, 67, 68, which are child segment nodes of the segment nodes 61, 64 typically form separate divisions within the departments, which are therefore not necessarily present in all cases. This is shown by the segment nodes, 62, 63, which do not include any child segments. However, if the segment nodes 61, 64 correspond to larger portions of the business, such as a subsidiary business, then the child segment nodes 65, 66, 67, 68 may represent departments within that subsidiary.

The principal nodes are then defined to correspond with individuals within the company or in external entities. In this case, each individual is represented by a respective principal entity, with each principal entity being associated with a respective principal node.

Similarly, object entities, which represent different objects, and application entities which represent different applications, are associated with respective application nodes.

Once the structure of the business has been determined in terms of the segment, principal, group and application nodes, then a model of the relationships between these nodes can be constructed using the policy server 20. In particular, the policy server 20 provides a graphical user interface (GUI), an example of which is shown in Figure 5, to aid this process.

The GUI shown in Figure 5 is associated with the policy database 36 by the policy server 20 so that as the structure of the business is defined using the GUI, the policy server 20 can create entries in the policy database so that the hierarchy structure is automatically reflected by the schema of the policy database 36.

In order to implement this, the policy database 36 includes a number of predefined tables, including a segment table, a principal table, a group table and an application table. Accordingly, a

respective table is provided for each type of node within the hierarchy model, allowing each of the tables to be used to define respective ones of the nodes that constitute the business hierarchy.

Thus, for example, each segment node is defined as a respective entry within the segment table. Accordingly, in object orientated terms, the segment table can be considered as representing a segment concept with each segment node being a respective instance of the segment concept.

The tables include pre-defined policy fields that are used to allow the policy of the business to be defined. This is achieved by providing data (usually in the form of a parameter or numerical value) to set the field so that it represents a particular policy.

The fields are typically defined as columns within the tables, with each node being provided on a separate row. Thus, for example, the segment table may include such fields as "password expiry duration", "password length", "password warning", "password age", and the like. An example of this structure is shown in Table 1 below.

Table 1

Segment	Password Expiry Duration	Password Length	Password Warning	Password Age

Accordingly, by entering data into the fields, this allows certain characteristics of users passwords to be defined at the segment level, as will be explained in more detail below. The database tables typically include default fields that allow a policy officer to define a security policy by simply entering appropriate data. However, the option is also available to allow the policy officer to define further fields, as required.

Data is entered into the tables using the GUI mentioned above. Accordingly, the policy officer is therefore able to define the segment nodes, 61, 62, 63, 64, which correspond to segments within the business by simply entering data representative of the respective segment.

An example of this is shown in Figure 5. This example is based on a fictional company XYZ, which includes offices in the USA and Asia.

As shown in Figure 5, the GUI includes a structure view 100, which shows the structure of the hierarchy which has so far been defined, and an editor screen 101 which is used to define the nodes and entities contained within the structure, as well as to edit policies for the defined structure nodes and entities.

Accordingly as shown in Figure 5 the policy officer has defined a root segment node 110, which represents the company XYZ.

The root segment node has two child segment nodes which have been defined, namely a USA office 111 and an Asia region 112, the latter of which has a further child segment node, namely a corporate segment node 114, as shown.

Each of the root segment node 110, the USA segment node 111, the Asia region segment node 112 and the corporate segment node 114 include respective principal nodes 120, 122, 123, 125 which are used to define the security policy for respective individuals within these respective segments.

In addition to this, the root segment node 110, the Asia region segment node 112 and the corporate segment node 114 all include respective group nodes 121, 124, 126, which allow policy to be set for groups of individuals.

Finally, the application node 113 includes a banking applications entity 127, an SSL certificate applications entity 128, an SSL normal applications entity 129, and a trading applications entity 130.

As mentioned above, as each node and each entity within the hierarchy is defined using the policy server, data is written into an appropriate table within the policy database.

Thus, as the USA office is defined as a segment node, an appropriate entry will be made in the segment table. The appearance of the segment table when all the segment nodes have been defined will be as shown in Table 2.

Table 2

Segment	Password Expiry Duration	Password Length	Password Warning	Password Age
XYZ Company				
USA Office				
Asia Region				
Corporate				

The policy server 20 then uses this information to generate the structure view 100, as shown in Figure 5.

As shown in the table 2 above, the segment table defines all the policy fields that are associated with the segment nodes. As would be appreciated by a person skilled in the art, these fields are predefined by the programs, although they can be adjusted by the policy officer if required.

As the system administrator defines the USA office segment node 111, the user will also define that this is related to XYZ company segment node 110, by a parent-child relationship. Accordingly, a link is defined between the USA Office entry and the corresponding XYZ Company in the segment table.

This is achieved by having a unique segment ID for each segment in the table, and having a parent ID column in the table, indicating the segment ID of any parent segments, as shown in Table 3, which shows a restricted portion of the segment table.

Table 3

Segment	Segment ID	Parent ID	Password Expiry Duration	Password Length
XYZ Company	110			
USA Office	111	110		
Asia Region	112	110		
Corporate	114	112		

Thus, for example, the USA office segment node includes the segment ID of the XYZ segment node in the Parent ID column, thus indicating the XYZ segment node is the parent of the USA Office

segment node. Similarly, the XYZ segment node does not include a segment ID in the parent ID column, indicating that it is the root segment node.

This process is then repeated for each of the nodes of the business hierarchy, so that all the nodes of the hierarchy are entered in the policy database in an appropriate one of the tables.

Thus, as would be appreciated by a person skilled in the art, the policy database 35 will include a segment table, a principal table, a group table and an application table. As the hierarchy model is defined, this is reflected by links between entries in the tables and hence is reflected by the database schema.

In order to define this structure, the user would utilize the edit screen 101. In its default state the editor screen includes a structure window 140, and policy window 141, and a registry window 142. In addition to this, an add button143, an edit button 144, a delete button 145, a save button 146, and a cancel button 147 are provided.

In use, if the user wants to add an additional node into the structure, the user need simply click on the add button 143. The user will then be prompted to enter the type of node which is to be added within the structure.

Once the type of node has been determined, the user will be prompted for additional information, which is used to uniquely define the node, and this includes the requirements for a name, which must be entered into a name field 148 and a brief description of the node which must be entered into a description field 149.

Once this has been done, the user can access the policy window 141 to define the policies for the respective node. As described above, a number of default policies are pre-defined within the policy database tables, and these are displayed as shown.

A policy propagation control 150 is provided which allows the user to select whether the security policy of the effective node is inherited from a parent node 151a or whether the policy is a new policy which overrides the policy of the parent as shown at 151b. In this case, if the policy is inherited, it will be identical to the policy of the parent node. In the present example, the XYZ Company root segment node is the parent node of the USA Office segment and accordingly, if the inherit box 151a is checked, the policy implemented by the USA office segment node will be identical to that implemented by the XYZ Company root segment node.

In this case, the data entered into the policy fields of the USA segment node, will be obtained directly from the respective XYZ company entry in the segment table of the policy database.

The policy server also includes a turn on maker checker field 151, which is used to activate a maker checker procedure which will be explained below, a password lock out field 161, which is used to lock out a user if they enter an incorrect password a predetermined number of times, and a password restriction field 152.

The password restriction field 152 allows the system administrator to define restrictions on the password, such as a password expiry duration 153, a character requirement for the password 154, a warning towards password expiry 155, a password age 156, a special character field 157, and a remember password section 158. In addition to this, an access time restriction 159, and an access location restriction location 160 are also defined.

Data reflecting the security policy defined by selecting appropriate entries for the fields is then entered into the respective portion of the respective tables in the policy database.

Thus, in the example of the segment table given above, the table would be modified as shown in Table 4 (please note that for clarity purposes only columns corresponding to some of the fields 150-160 have been shown, although in practical implementations a column corresponding to each field would be provided).

Table 4

Segment	Password Expiry Duration	Password Length	Password Warning	Password Age
XYZ Company	30 days	>0	3 days	1 day
USA Office	30 days	>0	3 days	1 day
Asia Region				
Corporate				

Similarly, policy fields are also defined for the remaining principal, group and application nodes. In general these fields will be similar to those used for the segment nodes.

Once the node have been defined, similar provisions are made for defining the principal, group, application and object entities. In this case, again policy settings are defined using the policy editor GUI.

In the case of the principals, an example of the edit screen 101 is shown in Figure 6. In this case, the edit screen 101 again includes a principal ID 170 identifying the respective principal, together with a name field 171 which includes the name of the principal and a distinguished name 172 which is one of the attributes of this user's X509 digital certificate.

The edit screen also includes an information window 173, which is used to provide basic information about the principal, such as their location in the business, or the like.

An accounts screen 174 is provided which sets out details of the principals user account, such as the amount of time the user has spent logged on, or the like.

A privilege screen 175 setting out any access privileges assigned to the user. This allows the security manager to define access to applications or information managed by segment nodes other than the users own. Thus, in this example, the user "Bob" may form part of the banking back-end applications entity, but may require access to applications or information which forms part of the home banking applications entity. Accordingly, the user "Bob" is given access privileges under the home banking applications entity.

In use, this is achieved by setting an access privilege for the principal entity "Bob" for the home banking and banking backend applications entities defined in the applications table within the policy database, as shown by the ticks 178, 179.

A memberships screen 176 is used to define which groups the user belongs to.

A policy screen 177 (not shown in detail) is used to define user specific policy, and will therefore set out the security settings which apply to the respective user. Again by default, an inherit option is available to allow the user to simply inherit policy settings from other entities or nodes. Alternatively, the

settings can be defined on an individual basis.

Once the policy settings have been defined for the entities, the data is written into an appropriate one of the database tables. Thus, for example, once the policy settings have been defined for a principal entity, a respective entry is made in the principal table, with the respective entity being entered in a respective row of the table.

A similar process is followed for group entities, allowing the access policies to be defined for a group of principal entities which are associated with the respective group entity. Thus, in the example above, if the principal entity "Bob" were part of a "Manager" group entity, then access privileges for the group entity "Manager" could be set to allow access to the home banking and banking backend application entities, in a manner similar to that described above with respect to the "Bob" principal entity, thereby allowing "Bob" the required access rights.

Similarly, the security policy for application and object entities is controlled by an application management screen as shown for example in Figure 7.

In this example, the structure view 100 shows that the banking applications entity 127 contains a number of object entities 190, such as a cusaccount object entity 191 which is currently selected.

The applications management screen includes an editor screen 101 which in turn includes a name field 180, a description field 181 and a type field 182, which are together used to identify the respective application and/or object entity which is currently selected. Thus, as shown the name field 180 include the name "cusaccount" of the customer account object entity.

Method, role, and permission screens 183, 184, and 185 are also provided. The method screen is used to provide details of the method by which is application is accessed, whereas the role screen 184 is used to define the role of the application.

Roles represent the users of application and their access permissions from application manager's perspective, that is, who is going to use the object of this application, what access rights they have to the object, and under what conditions. Roles are application-specific, e.g. Teller, Supervisor, Customer. Note that the Customer role of application A may be different from the Customer role of application B.

Finally the permission screen 185 sets the parameters required for a user to gain access to the application and/or object. Thus, for example, the permission screen 185 indicates access privileges required by a user to access the respective application, as well as any other additional criteria.

An example of the parameters that would be set for the cusaccount object are that in order to access object "cusaccount" using method "transfer", the user must have "customer_c2" role, must log on using "certificate", from "LAN", within timezone "0900-1600", the curreny must be "SGD" and the amount must be less than "1000"

As will be appreciated by persons skilled in the art the information required for the application management may be entered manually by hand, or through other methods e.g. uploaded, or alternately may be obtained by inheritance of information entered elsewhere within the GUI system. As a further alternative, software can be adapted to automatically add selected information, such as the name and description within the respective fields, as the software is installed on the network.

Accordingly, a policy officer, when presented with the GUI, can utilize the GUI to define the

policy for the business on a number of different levels. To summarize, this is achieved by first determining a hierarchy model which models the actual structure of the business. The hierarchy model is then imposed on the database schema by defining the hierarchy model using the policy server and the associated GUI, with links between various nodes in the defined hierarchy model being represented by links between entries within the database tables.

Once this has been completed, it is then possible for the policy officer to set the security policy, firstly for the root segment node, and then for each subsequent node and entity in the hierarchical model.

Typically by default, the policy applied to a higher level is automatically inherited by lower levels within the database, using for example, the inherit option 150a which forms part of the propagate policy control 150. However, it is possible at any stage to override this inheritance to provide alternative security policy features.

Thus for example it may be that the password requirement in the USA Office is far more restricted than for the Asia Region Offices. Accordingly, for these segment nodes which inherit policy from the root segment node, the policy officer need merely specify new policy settings for the segment node whose policy differs from that of the root segment node segment. However, the system can configured to prevent policy settings of lower segments from being weaker than those of the root segment.

As the policy settings are entered in the respective fields in the policy window, data representative of the entry is written into the database, as described above.

In the case of the principals, the data representative of the policy settings can be stored either in the policy database, as is the case for the policy settings of the segment nodes, or in the user database 35. In this case, the user database will include an entry for each principal entity, with the entry providing data representative of the respective user name, and password. In this case, the principal table in the policy database 36 will specify the security settings for the principal node, with an association being defined between the principal entity and the principal node. This association allows the policy of the principal entity to be defined by the settings selected for the respective principal node.

In use, once the security policy has been defined, the security policy is then implemented by the access server 23. In order to achieve this, when a user of one of the end stations 2 wishes to access material via the network 1, the user will be requested to enter their name and password or other credentials, either by the WSA (web security agent) which forms part of the web server 4, or an ASA (application security agent) which forms part of the applications server 6.

The access server 23 will then use the authentication daemon 24 to access the user database 35, via the registry API 34, to ensure the password and user name are valid. Assuming this to be the case, the access server 23 will cause the WSA or ASA to allow the user to have access to the network in accordance with the access privileges defined for that user.

Once this has been achieved, all subsequent operations that require security clearance must also be checked. As described above, this is achieved by having the access server 23 assign a session ID to each user whom is logged on to the network. Whenever an action is to be taken, the access server 23 will cause the authorization daemon 25 to access the policy database 36, via the registry API 34, to determine if the particular action is allowable. This check is performed by checking the policy settings for the respective

user, using the session ID to identify the user, and allowing the access server 23 to determine if the action should be allowed.

If the action is allowed, then an indication of this is transferred to either the WSA or ASA as appropriate, otherwise, an inhibit response is transferred causing the WSA or ASA to prevent the user taking the requested action.

The above discussion has focussed on the setting of security policy by a policy officer for principals, groups and applications of a business. The subsequent sections will focus on the administration aspect of the security system. In general, security administration, reviewing and enforcing the security policy settings for the entire business will be best achieved by dividing the responsibility amongst a number of system operatives.

Accordingly, it is normal to define a number of administration functions including policy officers, security auditors and security administrators.

The policy officers are in charge of defining the security policy for the entire business as a whole, using the present invention. The security auditors are in charge of controlling the generation of the audit files.

Finally, the security administrators are in charge of administering the security policy, application access permissions, user credentials, and privileges, at this is typically achieved by providing at least one security administrator for each segment, with the security administrator being responsible for security within the respective segment node and any associated child nodes.

In general, with the security administrators in charge of security within their own respective segment, the security administrators are provided with limited powers to alter policy, which will differ for each administrator of each segment. Accordingly, the administrators are assigned different rights at different segments, the rights assigned being only those rights which are required for their job. Thus, in the above example, a first security administrator may be associated with and have certain rights for the USA office segment node, whereas a second different security officer may be placed in charge of the Asia region segment node. In this case, the first security officer will typically have no rights over the Asia region segment node.

Examples of administration rights implemented are as follows:

Policy(POL):	Right to set and change security policy
Audit(AUD):	Right to review audit logs
Traverse(TRA):	Right to traverse the hierarchy
SegmentView(SVW):	Right to view (the details of) segment
PrincipalView(PVW):	Right to view (the details of) user
ObjectView(OVW):	Right to view (the details of) object
SegmentControl(SCN):	Right to create/delete/modify administrator and segment structure (User, Group, application nodes and sub-segment nodes)
PrincipalControl(PCN):	Right to create/delete/modify user
ObjectControl(OCN):	Right to create/delete/modify object

PrincipalEnable(PEN):	Right to enable/disable user
PrincipalForce(PFR):	Right to force user change password
PrincipalReset(PRS):	Right to reset user password
AdministratorEnable(AEN):	Right to enable/disable administrator
AdministratorForce(AFR):	Right to force administrator to alter password
AdministratorReset(ARS):	Right to reset administrator password

As mentioned above, each security administrator can usually only alter the policy settings of their own respective segment and any nodes such as group nodes, principal nodes or application nodes, or sub-segment nodes, which fall within the respective segment.

However, the rights given to the security administrators can be carefully controlled. Thus for example, the security administrators would typically not be given the POL administration rights which allows the security administrator to change the security policy within the segment. Instead they are typically only given rights such as PCN, OCN, PEN, or other similar rights, which allow them to control the access privileges of independent entities within the nodes.

Furthermore, the rights assigned to the security administrators are set for each segment separately. As a result, a security administrator may have far greater administration rights in one segment than another.

It will be appreciated from this that the rights for the security administrators must be set at the segment node level, using the policy editor GUI, to allow different rights to be provided for different security administrators in each segment.

As a result of this system, the overall security policy of the business can be maintained by the policy officer, whilst allowing the security applied to an individual to be changed by a respective security administrator, as necessary.

The policy officer will generally have the right to overturn any action taken by any security administrator. In particular, the policy officer is provided with an option which allows the synchronisation of the security policy for all segment nodes within the business. Accordingly, this can be used to ensure that a uniform security policy is in place across the entire business.

By allowing the administration rights for each security administrator to be tightly controlled, this prevents administrators being able either accidentally, or intentionally, to interfere with the security policy of the business, or access privileges of individuals in other segments.

In order to provide a further control, such powerful administration rights are subject to a security feature called maker-checker control.

This system operates by assigning at least two security administrators to a respective segment. In this case, any one of the security administrators with a POL Maker right has the right to propose changes to security policy, whereas a second security administrator with a POL Checker right only has the right to confirm changes proposed by the first security administrator. Accordingly, a change in policy can only be achieved if this is first proposed by the first security administrator and then confirmed by the second security administrator, thereby preventing one single individual having the right to modify security policy.

The present invention also provides a system for logging actions taken by users of the system.

This is achieved using the audit daemon 29, which is located within the security system 3. In use, the audit daemon 29 operates to record each action that is taken within an audit log table in the log store 38.

This operation can be achieved by having the server components, such as the authentication and authorization daemons 24, 25 call the audit daemon whenever an action is in progress. Accordingly, each time an authorization for an action is provided by the authorization daemon, this fact is recorded by the audit daemon 29 in a current log table. The audit daemon 29 will store an indication of the nature of the request (i.e. what action was requested) and whether the action was authorized (i.e. whether a proceed or inhibit response was generated by the access server 23).

Information regarding each action taken is written into the audit log table as a new record. This record will be digitally signed before it is written into the audit log table, with an indication of the time and date at which the action occurred. This provides integrity protection of the audit log, and allows the audit daemon 29 to access the audit log table at a later date and determine at what time a particular user performed a particular action.

The audit log table may periodically be stored or archived, with a further digital signature being provided during the process to ensure security of the records. This signature will include encoded information regarding the number of actions recorded and when the log was created. Typically this will occur either after a predetermined amount of time, such as one hour, or after a predetermined number of actions have been recorded and the log file is full.

In addition to this, the log store 38 could be located on a dedicated secure server, thereby further preventing the logs being tampered with.

Thus, the present invention operates to enforce enterprise security policies effectively by:

a) Defining security policies, user privileges and object permissions based on a segmented hierarchy model of users, groups and applications. A segment is an administration unit, which may consist of Users (Principals), Objects (Applications), and sub-segments. The segmented hierarchy is closely mirrors a company's existing organization structure. Roughly speaking, a segment corresponds to an organization's department or unit that manage its own applications and users.

b) The security policy can be defined at root segment level or sub-segment level. A sub-segment can inherit the security policy from its parent segment. Sub-segments may *override* its parent's policy. On the other hand, parent segments can *synchronize* its security policy within all subordinated segments. The change of security policy at a segment can immediately be enforced to whole segments and its subordinated segments.

c) Users and objects are defined within the segmented hierarchy. They are automatically subjected to the security policy specified at the segment where they are defined.

d) The security policy is defined by Security Policy Officers, used and enforced by Security Administrators, reviewed by Security Auditors.

To provide granular access control of security administration with built-in support for the best industrial security practices & principals like Segregation of Duty and Least Privilege.

a) Security administrators are segregated from system administrators through the use of different

administration interfaces. In particular, security administrators use the policy editor, whilst the system administrators use the service server.

b) Security Administrators are defined at segment level and can be assigned to very granular administration rights, as set out above.

c) Security Policy Officer, Security Auditor and Security Administrator functions are differentiated from each other by rights. Administrators are assigned to different rights at different segments. They can be only assigned to those rights, which are required for their job. Access Matrix can ensure that Policy and Audit will not assigned to those security administrators who have other administration rights.

d) The scope of the assigned administration *rights* is limited to the segment where this administrator is defined and its subordinated segments.

e) Administrators without the Policy right cannot modify security policies.

f) Administrator with *Delegation Option* can delegate part or all of his/her rights to other administrators. The grantor can later on revoke the rights he/she granted to a grantee, which cause root segment to revoke the rights that the grantor has granted, automatically and recursively.

g) Changes to powerful administration rights i.e. those with control rights can be subjected to *Maker-Checker* control. In a segment where the Maker-Checker control is on, all modification-related administrator's rights will be marked as either Maker or Checker. Only the Maker can submit requests, which must be approved by respective Checker before the root segment performs the action. Maker-Checker Control policy can be configurable to turn on or off at segment level.

h) Users are only assigned to the roles of those applications and objects, to which they are required to access. The present invention defines Mutually Exclusive Role Group (MERG) to ensure that a user at any time can assume at most one role within a MERG.

i) Dynamic Segregation of Duty within business applications is supported by parameter-conditions. For example, assume that withdraw cash needs to be duly controlled by two roles: Submitter and Approver and two methods: *Withdraw_Cash_Submit* and *Withdraw_Cash_Approve*. The following condition will ensure that the Approver is not the Submitter of same transaction:

<Approver,
Withdraw_Cash_Approve,
(Submitter != ROLE_HOLDER) >

(Here the Submitter is input parameter while ROLE_HOLDER is system-defined variable that is set to the current holder of the role i.e. Approver. Note that same person may be assigned to these two roles because of small team sizes.)

Accordingly, the present invention seeks to provide a security administration framework for large-

size, mission-critical and security-sensitive environment which provides:

1) Built-in support for enterprise-wide security policy enforcement.
2) Built-in support for best industrial security practices & principals like Segregation of Duty and Least Privilege
3) Scalable to support many different users and applications
4) Flexible enough to define and control security-related business rules within one single organization or across multiple organizations.

It will be appreciated by persons skilled in the art that numerous variations and modifications will become apparent. All such variations and modifications which become apparent to persons skilled in the art, should be considered to fall within the spirit and scope of the invention as broadly hereinbefore described.

THE CLAIMS:

1) A hierarchy model for modelling the security policy of a business, the hierarchy model including:
   a) A number of segment nodes, each segment node being used to define the security policy which applies to a respective segment of the business, each segment being associated with a number of respective individuals and/or applications, each segment node including at least one of the following:
      i) A principal node, the principal nodes being used to define the security policy which applies to respective individuals within the business, each principal node including one or more principal entities, and each principal entity corresponding to a respective individual;
      ii) An application node, the application nodes being used to define the security policy which applies to respective applications or objects within the business, each application node including one or more application or object entities, and each application or object entity corresponding to a respective application or object;
      iii) A group node, the group nodes being used to define the security policy which applies to respective groups of principals within the business, each group node including one or more group entities, and each group entity corresponding to a group of respective individuals;
   b) A number of connections, the connections defining associations between the nodes to reflect the structure of the business.
2) A hierarchy model according to claim 1, wherein each segment node may include one or more child segment nodes, each child segment node defining the security policy for a logical unit within the respective parent segment.
3) A hierarchy model according to claim 2, wherein the hierarchy model further includes at least one root segment node, the root segment node being a segment node with no parent nodes.
4) A hierarchy model according to any of the preceding claims, wherein the model further includes application collection entities, each application collection entity representing a collection of application entities.
5) A hierarchy model according to any of the preceding claims, wherein the model is implemented within a database, the database including a number of tables, each table representing a respective type of node within the model.
6) A hierarchy model according to claim 5, the database including a segment table, and wherein each segment node is represented by a respective entry within the segment table.
7) A hierarchy model according to claim 5 or claim 6, the database including a principal table, and wherein each principal entity is represented by a respective entry within the principal table.
8) A hierarchy model according to any of claims 5 to 7, the database including a group table, and wherein each group entity is represented by a respective entry within the group table.
9) A hierarchy model according to any of claims 4 to 7, the database including a application table, each application entity, object entity and application collection entity being represented by a respective entry in the application table.

10) A hierarchy model according to any of claims 5 to 9, wherein the number of connections are represented by links between entries in the tables.

11) A hierarchy model according to any of the preceding claims, wherein each node within the model has an associated number of security settings, the security policy being defined by selecting appropriate ones of the security settings.

12) A hierarchy model according to claim 11, when dependent on any of claims 5 to 10, wherein each entry is defined in a respective row of the respective table, with each security setting being defined within a respective column.

13) A hierarchy model according to claim 11 or claim 12, wherein each type of node has respective security settings.

14) A hierarchy model according to any of the preceding claims, wherein each segment corresponds to a group of related business attributes capable of managing the associated applications and individuals.

15) A hierarchy model for modelling the security policy of a business substantially as hereinbefore described.

16) A method of generating a hierarchy model for modelling the security policy of a business, the method including:

a) Defining the business as a number of segments, each segment being a respective unit within the business;

b) Determining the individuals, groups of individuals and/or applications associated with each unit;

c) Defining a number of segment nodes, each segment node being used to define the security policy which applies to a respective segment of the business;

d) Defining at least one principal, group and/or application node, each principal, group and/or application node being used to define the security policy which applies to respective individuals, groups of individuals and/or applications; and,

e) Defining a number of connections between the defined nodes in accordance with the relationships between the segments, individuals and/or applications to reflect the structure of the business.

17) A method according to claim 16, the method further involving defining:

a) principal entities, each principal entity corresponding to a respective individual;

b) one or more application or object entities, and each application or object entity corresponding to a respective application or object; and,

c) one or more group entities, and each group entity corresponding to a group of respective individuals;

18) A method according to claim 16 or claim 17, wherein the method further includes:

a) Generating a database, the database being formed from a number of tables; and,

b) Associating each table with a respective type of node.

19) A method according to claim 18, wherein the method further includes defining:

a) Each segment node as a respective entry within the segment table;

b) Each principal entity as a respective entry within the principal table;

c) Each group entity as a respective entry within the group table; and,

d) Each application entity, object entity and/or application collection entity as a respective entry in the application table.

20) A method according to claim 19, wherein the method further involves defining the number of connections by defining links between entries in the tables.

21) A method of generating a hierarchy model substantially as hereinbefore described.

22) A method according to any of claims 16 to 21, wherein the hierarchy model is a model according to any of claims 1 to 15.

23) A computer program product including computer executable program code for generating a hierarchy model in accordance with the method of any of claims 16 to 22.

24) A computer program product according to claim 23, the computer executable code being stored on a computer readable medium.

25) A computer program product including computer executable program code for generating a hierarchy model substantially as hereinbefore described.

26) A system for generating a hierarchy model for modelling the security policy of a business, the hierarchy model including:

a) An input, adapted to receive inputs:

i) Defining the business as a number of segments, each segment being a respective unit within the business;

ii) Identifying the individuals and/or applications associated with each unit;

iii) Defining a number of segment nodes, each segment node being used to define the security policy which applies to a respective segment of the business;

iv) Defining at least one principal, group and/or application node, each principal, group and/or application node being used to define the security policy which applies to respective individuals, groups of individuals and/or applications; and,

v) Defining a number of connections between the defined nodes in accordance with the relationships between the segments, individuals and/or applications to reflect the structure of the business.

b) A store; and,

c) A processor adapted to:

i) Generate a number of tables in the store;

ii) Associate each table with a respective node in the hierarchy model; and,

iii) Define a number of links between the tables, each link representing a connection between respective nodes within the tables.

27) A system according to claim 26, wherein input is adapted to allow the user to define:

a) principal entities, each principal entity corresponding to a respective individual;

b) one or more application or object entities, and each application or object entity corresponding to a respective application or object; and,

c) one or more group entities, and each group entity corresponding to a group of respective individuals;

28) A system according to claim 27, wherein the processor is further adapted to define:

i) Each segment node as a respective entry within the segment table;

ii) Each principal entity as a respective entry within the principal table;

iii) Each group entity as a respective entry within the group table; and,

iv) Each application entity, object entity and/or application collection entity as a respective entry in the application table.

29) A system according to claim 28, wherein the processor is further adapted to define the number of connections by defining links between entries in the tables.

30) A system according to any of claims 26 to 29, the system further including a display, and wherein the processor is adapted to display icons to the user of the system, the icons representing segment nodes, application nodes, group nodes and principal nodes, the system being adapted to allow the user to define:

i) Define the business as a number of segments, by arranging segment icons on the display;

ii) Identify the individuals and/or applications associated with each segment by arranging application, group and principal node icons on the display; and,

iii) Define relative interconnections between the defined nodes by allowing the user to link the respective icons.

31) A system for generating a hierarchy model substantially as hereinbefore described with reference to the accompanying drawings.

32) A system according to any of claims 26 to 31, the system generating a hierarchy model according to any of claims 1 to 15.

33) A method of defining a security policy for a business, the method including:

a) Defining the business in terms of a hierarchy model, the hierarchy model including:

i) A number of segment nodes, each segment node being used to define the security policy which applies to a respective segment of the business;

ii) At least one principal, group and/or application node, each principal, group and/or application node being used to define the security policy which applies to respective individuals, groups of individuals and/or applications; and,

iii) A number of connections between the defined nodes, the connections representing the relationships between the segments, individuals and/or applications to reflect the structure of the business.

b) Specifying the security policy for at least some of the nodes in the hierarchy model; and,

c) Propagating the specified security policy to other nodes in the hierarchy model.

34) A method according to claim 33, wherein the hierarchy model is a model according to any of claims 1 to 15.

35) A method according to claim 33 or claim 34, wherein the method of defining the business in terms of a hierarchy model includes generating a hierarchy model in accordance with the method of any of claims 16 to 22.

36) A method according to claim 35, when dependent on claim 20, wherein the method of defining the security settings for a node includes:

a) Generating security data defining security settings for the respective node; and,

b) Inputting the security data as part of the respective entry within the respective table within the database.

37) A method according to claim 36, wherein the method further involves propagating the defined security settings to other nodes in the hierarchy model by propagating the security data to other linked entries in the database.

38) A method according to claim 30, wherein the security settings applied to a child segment node may override the security settings propagated from the respective parent segment node.

39) A method of defining a security policy for a business substantially as hereinbefore described.

40) A computer program product including computer executable code for defining a security policy for a business in accordance with the method of any of claims 33 to 40.

41) A computer program product according to claim 40 stored on a computer readable medium.

42) A computer program product including computer executable code for defining a security policy for a business substantially as hereinbefore described.

43) A system for defining a security policy for a business or extended organisations, the system including:

a) A system for defining the business in terms of a hierarchy model, the hierarchy model including:

i) A number of segment nodes, each segment node being used to define the security policy which applies to a respective segment of the business or extended organisations;

ii) At least one principal, group and/or application node, each principal, group and/or application node being used to define the security policy which applies to respective individuals, groups of individuals and/or applications; and,

iii) A number of connections between the defined nodes, the connections representing the relationships between the segments, individuals and/or applications to reflect the structure of the business.

b) An input, adapted to receive inputs specifying the security policy for at least one of the nodes in the hierarchy model; and,

c) A store for storing the specified security policy; and,

d) A processor adapted to propagate the specified security policy to other nodes in the hierarchy model.

44) A system according to claim 43, wherein the system for defining the business in terms of a hierarchy model, includes a system according to any of claims 26 to 32.

45) A system according to claim 44, when dependent on at least claim 28, wherein the system is adapted to receive inputs specifying the security policy for at least one of the nodes in the hierarchy model by receiving security data defining security settings for the respective node, the processor being adapted to input the security data as part of the respective entry within the respective table within the database.

46) A system according to claim 45, when dependent on claim 29, wherein the method further involves propagating the defined security settings to other nodes in the hierarchy model by propagating the security data to other linked entries in the database.

47) A system for defining a security policy for a business substantially as hereinbefore described with reference to the accompanying drawings.

48) A method of implementing a security policy for a business, the method including:

a) Defining a security policy for the business, the security policy being defined in terms of a hierarchy model including:

i) A number of segment nodes, each segment node representing a respective segment of the business;

ii) At least one principal, group and/or application node, each principal, group and/or application node representing a respective individual, group of individuals and/or applications within the business; and,

iii) A number of connections, the connections defining the associations between the nodes, the security policy being defined by security settings associated with each node;

b) Monitoring action to be taken by a respective segment, individual or application within the business;

c) Determining if the action is permissible in accordance with the defined security settings; and,

d) Allowing or preventing the action in accordance with the result of the determination.

49) A method according to claim 48, wherein the security policy is defined in accordance with the method of any of claims 33 to 39.

50) A method according to claim 48 or claim 49, wherein the security settings are stored in a database associated with the respective segment, principal, group and/or object node, the method of determining if the action is permissible including:

a) Accessing the database to determine the security settings for the respective segment, individual and/or application; and,

b) Comparing the accessed security settings to the action to be taken to determine if the action is permitted.

51) A method according to any of claims 48 to 50, wherein the hierarchy model is a model according to any of claims 1 to 15.

52) A method of implementing a security policy for a business substantially as hereinbefore described.

53) A computer program product including computer executable code for implementing a security policy in accordance with the method of any of claims 48 to 52.

54) A computer program according to claim 53 stored on a computer readable medium.

55) A computer program product including computer executable code for implementing a security policy for a business substantially as hereinbefore described.

56) A system for implementing a security policy for a business, the security policy being defined in terms of a hierarchy model, the hierarchy model including:

57) A system for implementing a security policy for a business, the system including:

a) A system for defining a security policy for the business, the security policy being defined in terms of a hierarchy model including:
   i) A number of segment nodes, each segment node representing a respective segment of the business;
   ii) At least one principal, group and/or application node, each principal, group and/or application node representing a respective individual, group of individuals and/or applications within the business; and,
   iii) A number of connections, the connections defining the associations between the nodes, the security policy being defined by security settings associated with each node; and,
b) A processor, the processor being adapted to:
   i) Monitor action to be taken by a respective segment, individual or application within the business;
   ii) Determine if the action is permissible in accordance with the defined security settings; and,
   iii) Allow or prevent the action in accordance with the result of the determination.

58) A system according to claim 57, wherein the system for defining a security policy for the business is a system according to any of claims 43 to 47.

59) A system according to claim 57 or claim 58, wherein the system is adapted to receive action data representing the action to be taken, and wherein the security settings are defined in terms of security data stored in a store, and wherein the processor is adapted to:
   a) Access the store to determine the security data for the respective segment, individual and/or application; and,
   b) Comparing the accessed security data to the action data to determine if the action is permitted.

60) A system for implementing a security policy for a business substantially as hereinbefore described with reference to the accompanying drawings.

61) A method of controlling the administration of the security policy of a business, the security policy being defined using a hierarchy model including:
   a) A number of segment nodes, each segment node having an associated number of security settings, the segment node being used to define the security policy which applies to a respective segment of the business by selecting appropriate ones of the security settings; and,
   b) A number of connection, the connections defining associations between the segment nodes to reflect the structure of the business, the method including:
      i) Assigning an administrator to each segment;
      ii) Defining administration rights that can be assigned to any administrator, the administration rights controlling the administration procedures the respective administrator can perform; and,
      iii) Assigning appropriate ones of the administration rights to the administrator, if required.

62) A method according to claim 61, wherein the method of assigning the administrator includes defining an administrator entity for each segment, the administrator entity representing a respective individual within the business.

63) A method according to claim 62, wherein the method of controlling the administration rights includes assigning appropriate ones of the administration rights to the defined administrator entity.

64) A method according to claim 63, wherein the administration rights are defined at the system level but the scope of the administration rights is limited to the segment where the administrator is defined and its subordinate segments.

65) A method of controlling the administration of the security policy of a business substantially as hereinbefore described.

66) A computer program product for controlling the administration of the security policy of a business, the computer program product including computer executable code for performing the method of any of claims 61 to 65.

67) A computer program product according to claim 66 stored on a computer readable medium.

68) A computer program product for controlling the administration of the security policy of a business substantially as hereinbefore described.

69) A system for controlling the administration of the security policy of a business, the security policy being defined using a hierarchy model including:

a) A number of segment nodes, each segment node having an associated number of security settings, the segment node being used to define the security policy which applies to a respective segment of the business by selecting appropriate ones of the security settings; and,

b) A number of connections, the connections defining associations between the segment nodes to reflect the structure of the business, the system including:

i) An input for:

(1) Receiving an indication of an administrator assigned to each segment; and,

(2) Defining administration rights; and,

(3) Assigning appropriate ones of the administration rights to the administrator, if required; and,

ii) A processor adapted to control the administration procedures the respective administrator can perform in accordance with the defined administration rights.

70) A system for controlling the administration of the security policy of a business substantially as hereinbefore described with reference to the accompanying drawings.

71) A method of controlling the administration of the security policy of a business, the security policy being defined using security settings, the method involving:

a) Assigning first and second administrators to control alteration of the security settings;

b) Controlling the first administrator to only allow the first administrator to propose changes to the security settings; and,

c) Controlling the second administrator to only allow the second administrator to accept or reject proposed changes to the security settings, thereby allowing the first and second administrators to control the security settings.

72) A method according to claim 71, wherein the method of assigning the first and second administrators includes defining first and second administrator entities, each administrator entity being associated with a respective individual within the business.

73) A method according to claim 72, wherein the method of controlling the administrators includes:

i) Defining a number of administration rights that can be assigned to any administrator entity, the administration rights controlling the administration procedures the respective administrator can perform; and,

ii) Assigning appropriate ones of the administration rights to the respective administrator entity, if required.

74) A method according to claim 73, wherein the administration rights are selected ones of the security settings and wherein the administration rights are selected to prevent an administrator altering the administration rights of their own respective administrator entity.

75) A method according to any of claims 71 to 74, wherein the hierarchy model includes:

a) A number of segment nodes, each segment node having an associated number of security settings, the segment node being used to define the security policy which applies to a respective segment of the business by selecting appropriate ones of the security settings; and,

b) A number of connections, the connections defining associations between the segment nodes to reflect the structure of the business, and wherein the method includes assigning respective first and second administrators to each segment.

76) A method for controlling the security policy of a business substantially as hereinbefore described.

77) A computer program product including computer executable code for controlling the administration of the security policy of a business in accordance with the method of any of claims 71 to 76.

78) A computer program product according to claim 77 stored on a computer readable medium.

79) A system for controlling the administration of the security policy of a business, the security policy being defined using security settings, the system including:

a) An input for assigning first and second administrators to control alteration of the security settings; and

b) A processor adapted to:

i) Control the first administrator to only allow the first administrator to propose changes to the security settings; and,

ii) Control the second administrator to only allow the second administrator to accept or reject proposed changes to the security settings, thereby allowing the first and second administrators to control the security settings.

80) A system for controlling the administration of the security policy of a business substantially as hereinbefore described with reference to the accompanying drawings.

81) A method of controlling the administration of the security policy of a business, the security policy being defined using a hierarchy model including:

a) A number of segment nodes, each segment node having an associated number of security settings, the segment node being used to define the security policy which applies to a respective segment of the business by selecting appropriate ones of the security settings; and,

b) A number of connections, the connections defining associations between the segment nodes to reflect the structure of the business, the method including:

i) Assigning an administrator to each segment, the administrator being responsible for the security settings of the respective segment; and,

ii) Allowing a first administrator to delegate responsibility for the security settings of a respective segment to a second administrator.

82) A method according to claim 81, wherein the method of assigning the administrator includes defining an administrator entity for each segment, the administrator entity representing a respective individual within the business.

83) A method according to claim 82, wherein an administrator is assigned to a segment by specifying the respective administrator entity in the security settings for the segment, and wherein the method of delegating responsibility includes specifying the second administrator in the respective security settings.

84) A method according to any of claims 81 to 83, wherein the method further includes defining a number of administration rights for each administrator entity, the administration rights defining which security settings can be altered by the respective administrator.

85) A method according to claim 84, wherein the method of delegating responsibility includes:

a) Determining if the administration rights for the first and second administrators allow responsibility to be delegated; and,

b) Selecting some or all of the administration rights for the second administrator to correspond to the administration rights for the first administrator, thereby allowing the second administrator to perform some or all of function of the first.

86) A method of controlling the administration of the security policy of a business substantially as hereinbefore described.

87) A computer program product including computer executable code for controlling the administration of the security policy of a business in accordance with the method of any of claims 81 to 86.

88) A computer program product according to claim 87 stored on a computer readable medium.

89) A computer program product including computer executable code for controlling the administration of the security policy of a business substantially as hereinbefore described.

90) A system for controlling the administration of the security policy of a business, the security policy being defined using a hierarchy model including:

a) A number of segment nodes, each segment node having an associated number of security settings, the segment node being used to define the security policy which applies to a respective segment of the business by selecting appropriate ones of the security settings; and,

b) A number of connections, the connections defining associations between the segment nodes to reflect the structure of the business, the system including:

i) An input for assigning an administrator to each segment, the administrator being responsible for the security settings of the respective segment; and,

ii) A processor adapted to allow a first administrator to delegate responsibility for the security settings of a respective segment to a second administrator.

91) A system for controlling the administration of the security policy of a business substantially as hereinbefore described.

92) A method of monitoring a security system which implements a security server, the method including:

a) Causing the security server to generate an indication of each action taken on the security system;

b) Digitally signing the indication of each action; and,

c) Storing the digitally signed indication.

93) A method according to claim 92, wherein the method includes storing the digitally signed indication in a table, each digitally signed indication being stored in a respective portion of the table.

94) A method according to claim 93, wherein the table is located remotely to the security system on ore or more dedicated systems.

95) A method according to claim 92, wherein the security server is adapted to generate an authorisation or rejection indication in response to a request to perform an action, and wherein the method further includes causing the security server to generate an indication of each authorisation or rejection.

96) A method of monitoring a security system substantially as hereinbefore described.

97) A computer program product including computer executable code for monitoring a security system in accordance with the method of any of claims 92 to 96.

98) A computer program product including computer executable code for monitoring a the each action taken on the security system substantially as hereinbefore described.

99) A system for monitoring a security system which implements a security server, the system comprising:

a) An input for receiving an indication of each action taken on the security system;

b) A processor for digitally signing the indication; and,

c) A store for storing the digitally signed indication.

100) A system according to claim 99, wherein the store is located remotely to the security system on one or more dedicated systems

101) A system according to claim 99 or claim 100, wherein the processor is adapted to store the digitally signed indication in a table, each digitally signed indication being stored in a respective portion of the table.

102) A system according to any of claims 99 to 101, wherein security server forms part of the system, and wherein the security server comprises a security processor adapted to generate an authorisation or rejection indication in response to a request to perform an action.

103) A system for monitoring a security system substantially as hereinbefore described.

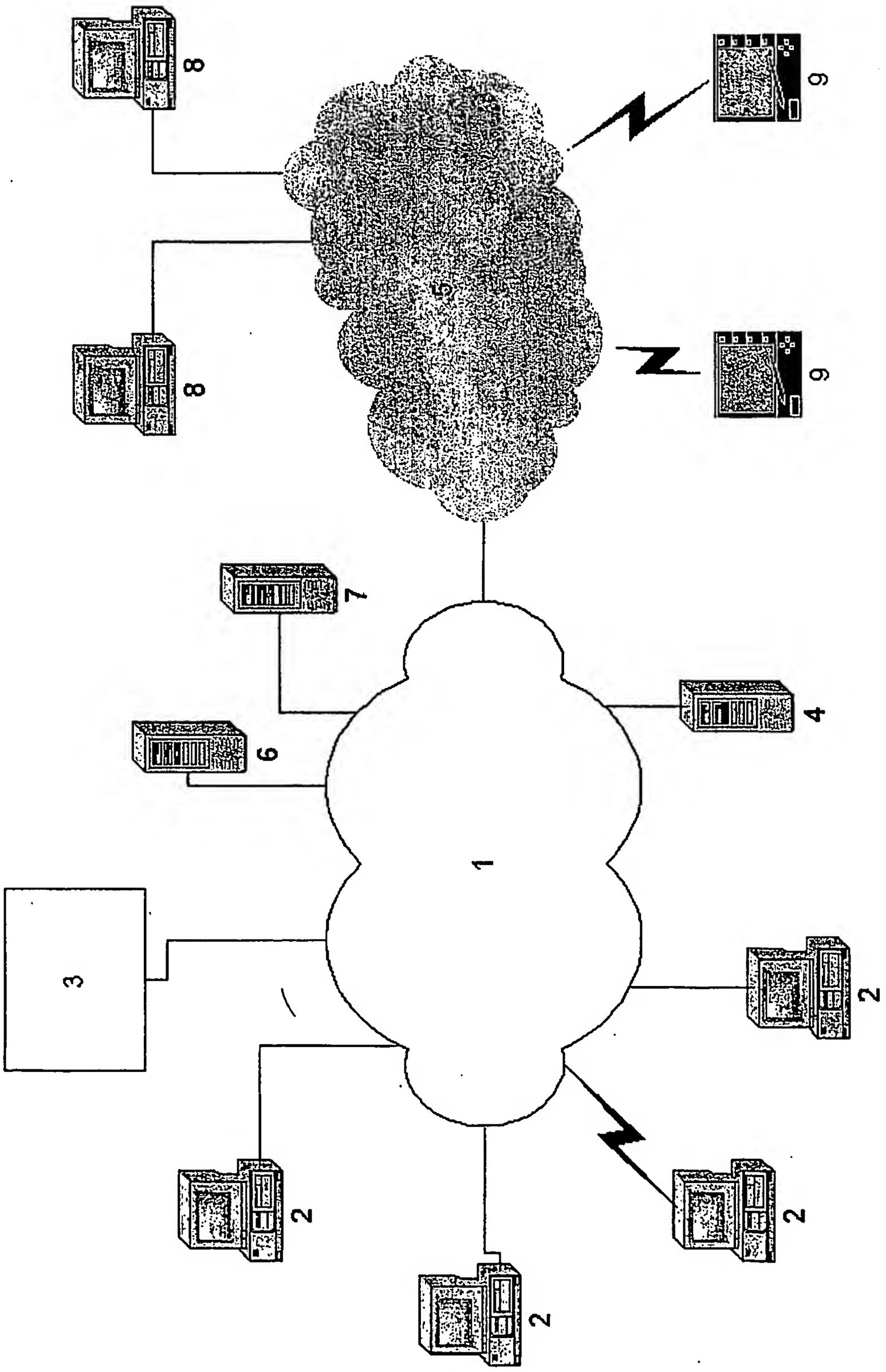


Fig. 1

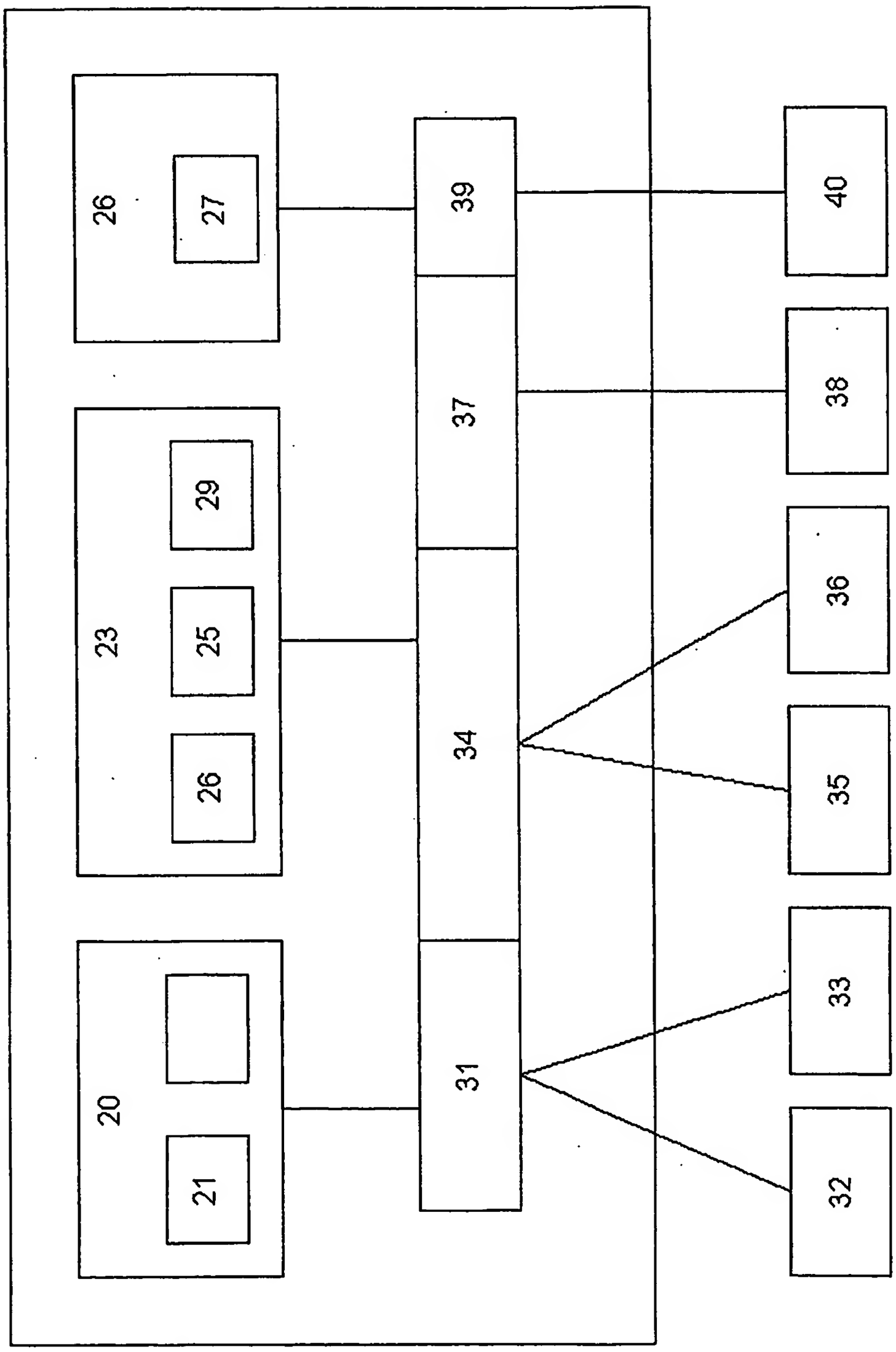


Fig. 2

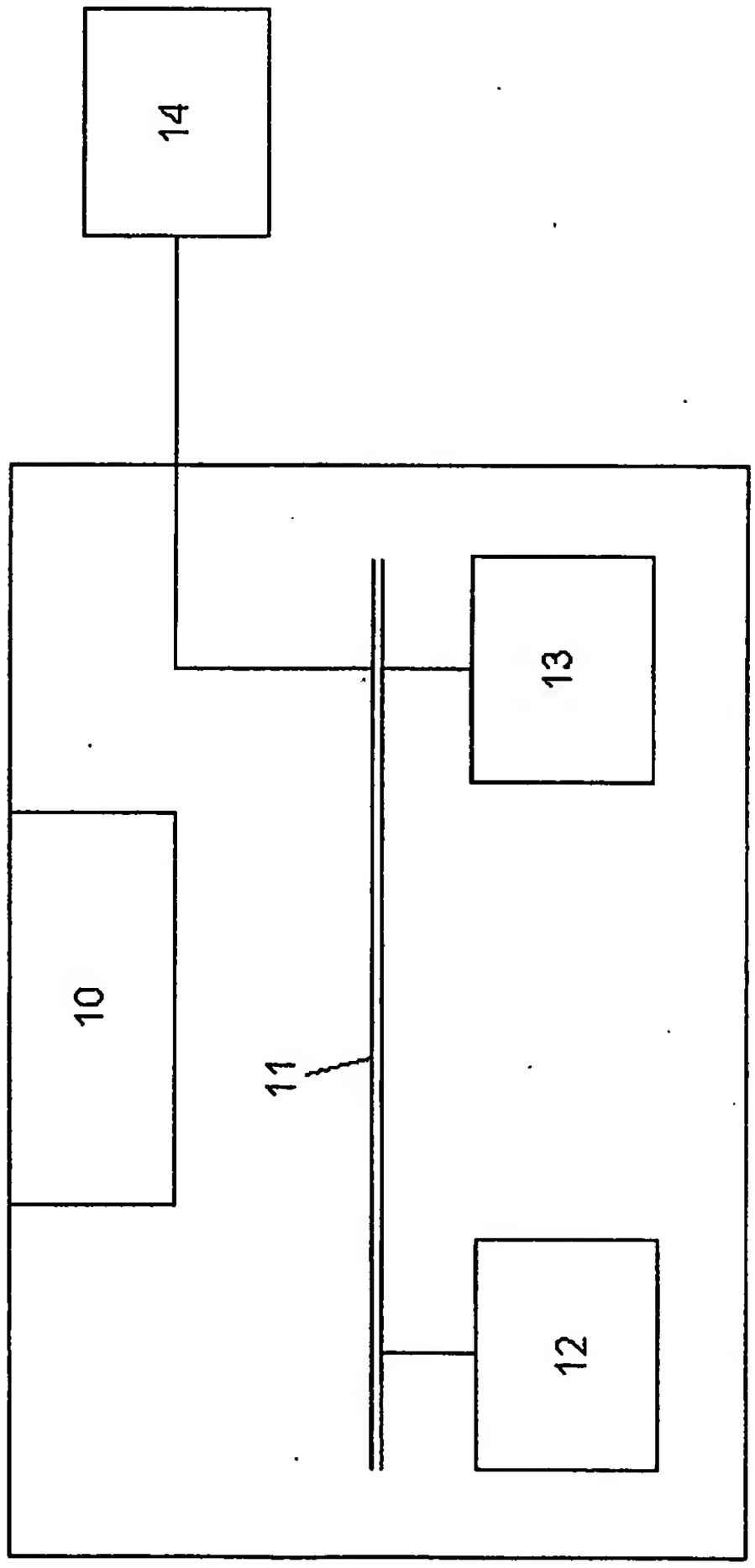


Fig. 3

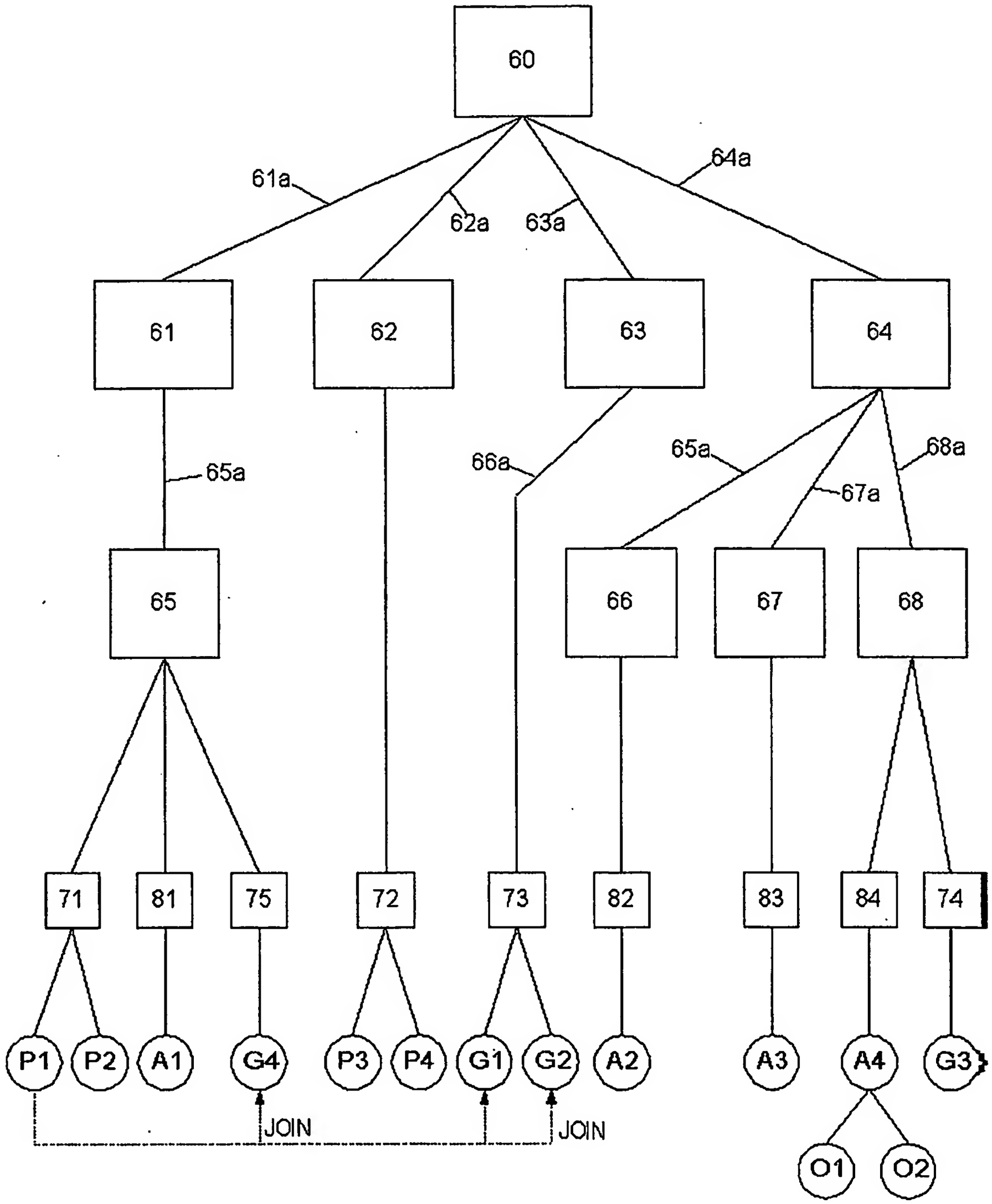

60
61a
62a
63a
64a
61
62
63
64
65a
66a
65a
67a
68a
65
66
67
68
71
81
75
72
73
82
83
84
74
P1
P2
A1
G4
P3
P4
G1
G2
A2
A3
A4
G3
JOIN
JOIN
O1
O2

Fig. 4

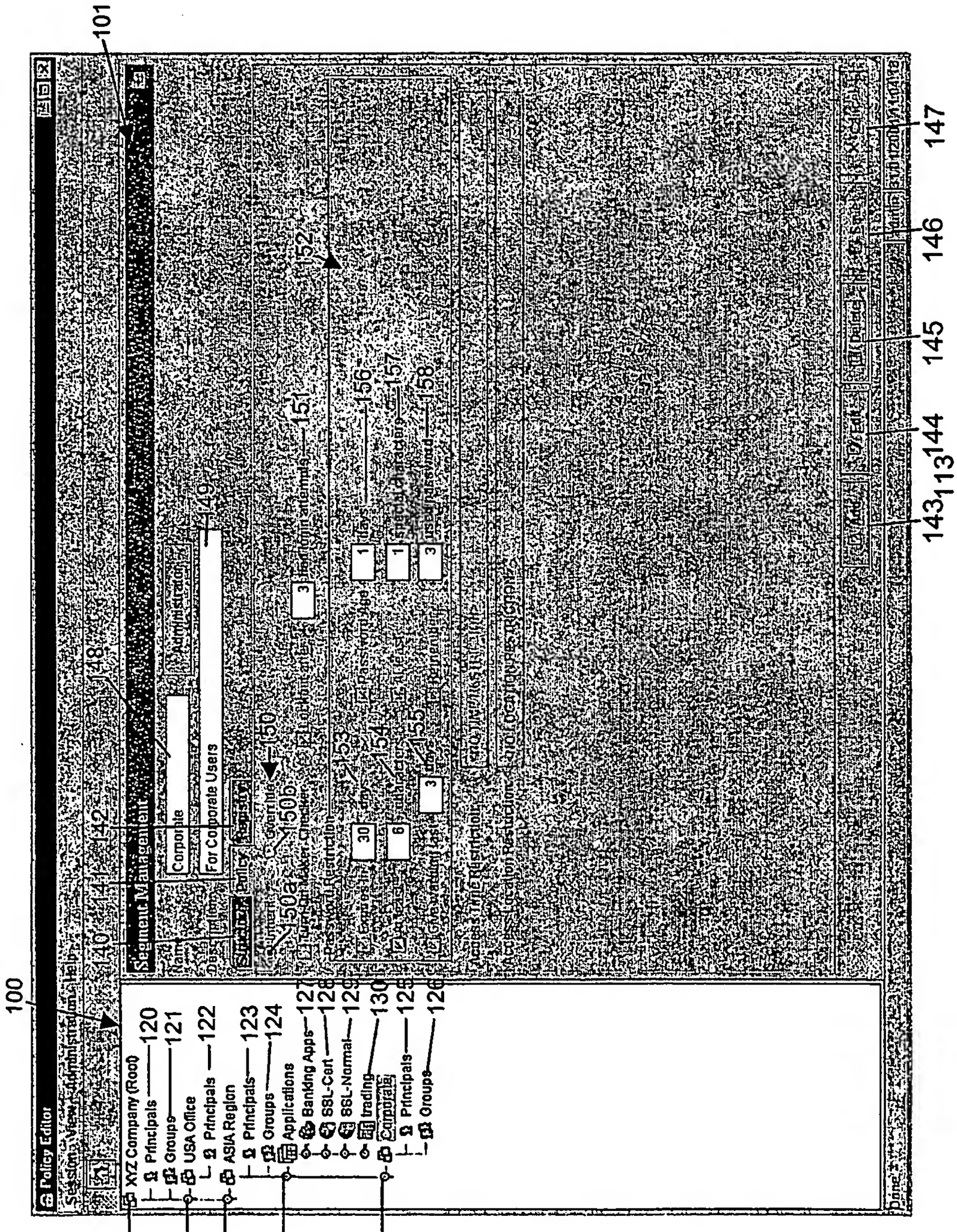

101
147
146
145
144
143
113
100
Policy Editor
Segment Management
XYZ Company (Root)
Principals
Groups
USA Office
Principals
ASIA Region
Principals
Groups
Applications
Banking Apps
SSL-Cert
SSL-Normal
trading
Corporate
Principals
Groups
110
111
112
113
114
120
121
122
123
124
125
126
127
128
129
130

Fig. 5

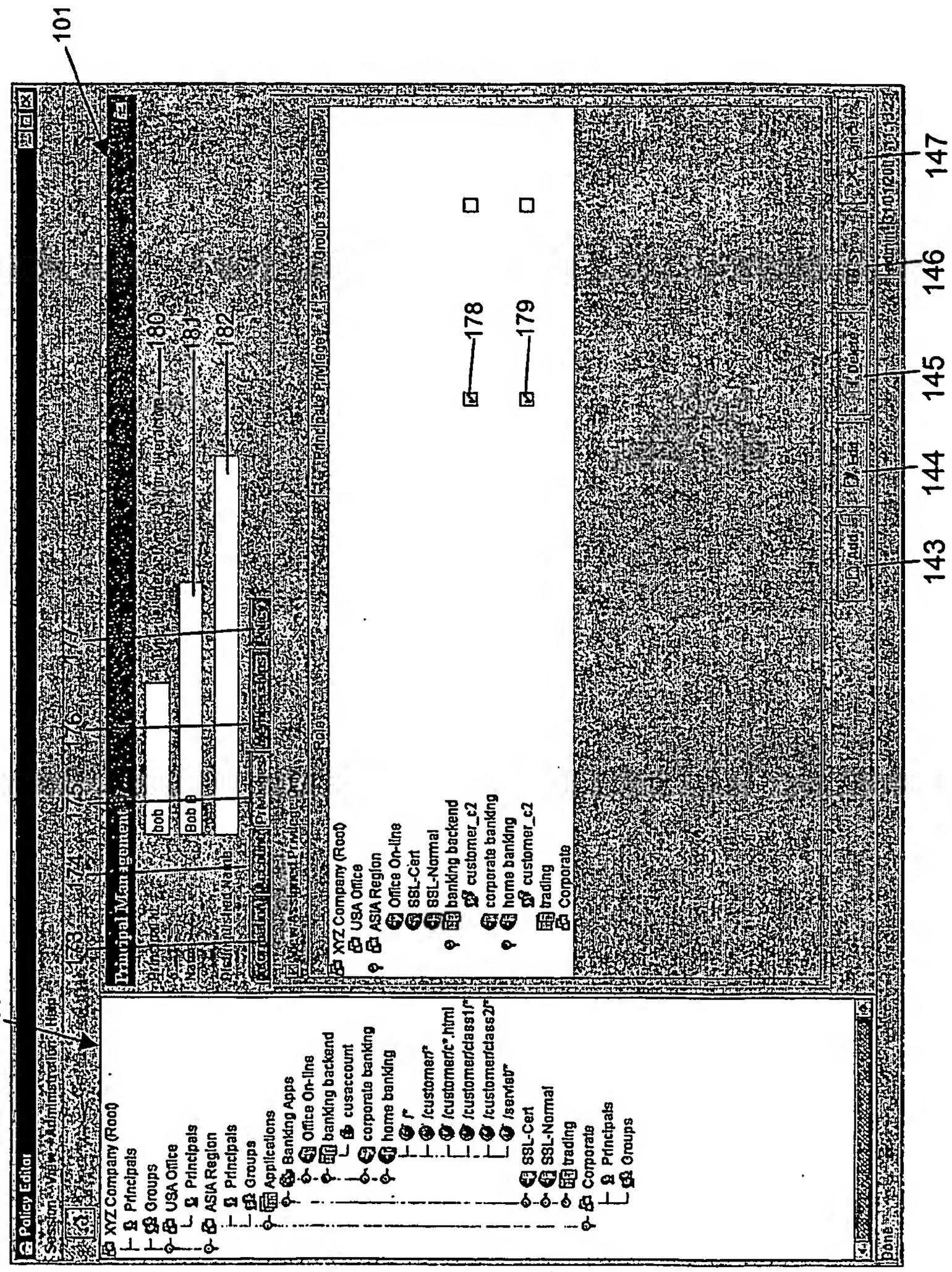

100
101
147
146
145
144
143
178
179
180
181
182
Policy Editor
XYZ Company (Root)
Principals
Groups
USA Office
Principals
ASIA Region
Principals
Groups
Applications
Banking Apps
Office On-line
banking backend
cusaccount
corporate banking
home banking
/
/customer*
/customer/c*.html
/customer/class1/*
/customer/class2/*
/servlet*
SSL-Cert
SSL-Normal
trading
Corporate
Principals
Groups

Fig. 6

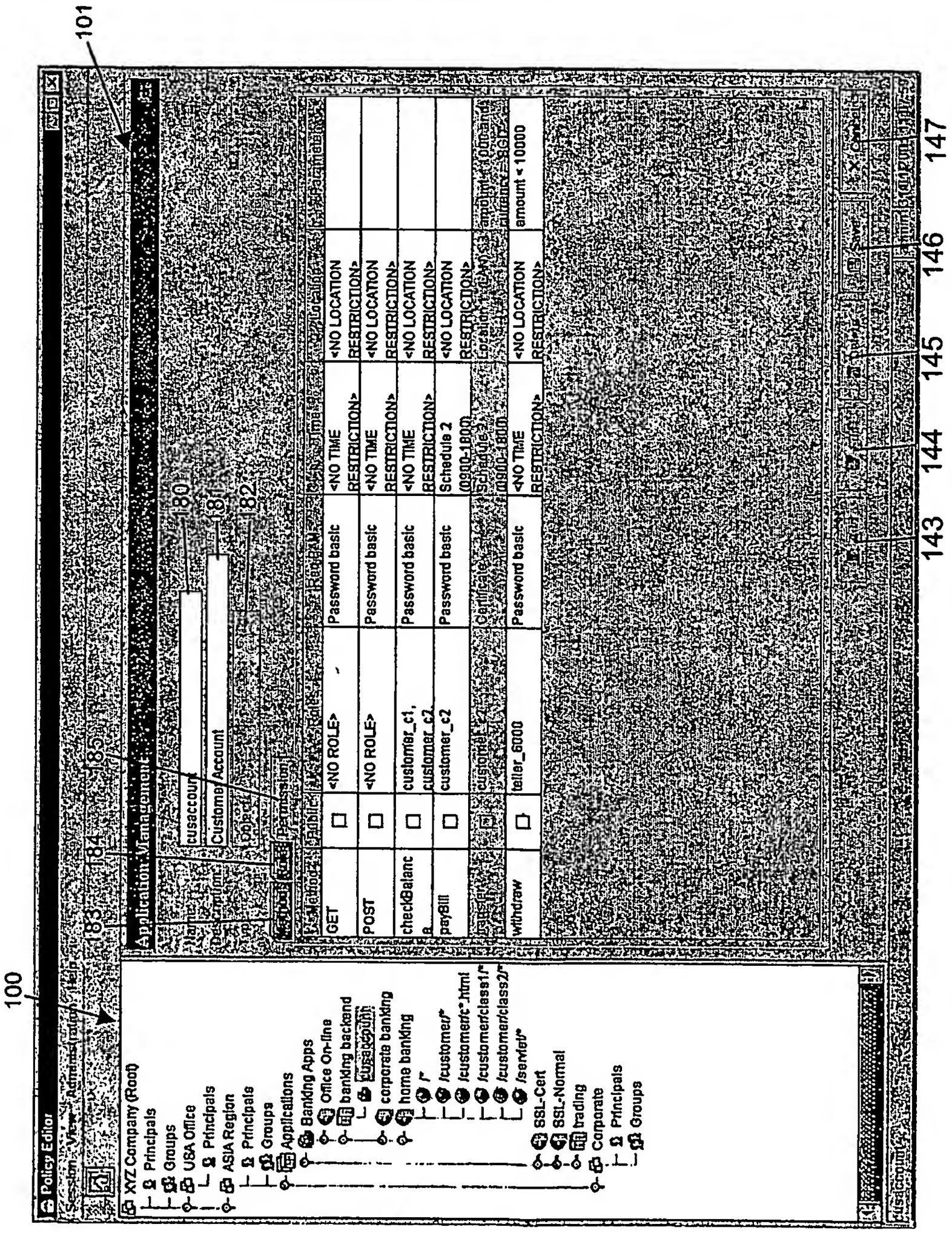

100
101
147
146
145
144
143
180
181
182
183
184
185
Policy Editor
XYZ Company (Root)
Principals
Groups
USA Office
Principals
ASIA Region
Principals
Groups
Applications
Banking Apps
Office On-line
banking backend
corporate banking
home banking
/
/customer/*
/customer/c*.html
/customer/class1/*
/customer/class2/*
/servlet/*
SSL-Cert
SSL-Normal
trading
Corporate
Principals
Groups
Application: Management
Name:
Description:
Type:
Methods
Public
Permissions
Method Name
Password basic
Certificate
Password basic
GET
POST
checkBalanc
payBill
withdraw
<NO ROLE>
customer_c1, customer_c2
customer_c2
customer_c2
teller_6000
<NO TIME RESTRICTION>
Schedule 2 (0900-1800)
<NO LOCATION RESTRICTION>
amount < 10000

Fig. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SG02/00027

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl. [7]: G06F 17/60, H04L 12/24

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
WPAT, USPTO, (model, hierarchy, policy, security, administration)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6161139 A (Win et al) 12 December 2000 Whole Document	1-103
X	US 6023765 A (Kuhn) 8 February 2000 Whole Document	1-103
X	US 5797128 A (Birnbaum) 18 August 1998 Whole Document	1-103

☐ Further documents are listed in the continuation of Box C ☒ See patent family annex

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
5 July 2002	16 JUL 2002
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA E-mail address: pct@ipaustralia.gov.au Facsimile No. (02) 6285 3929	Authorized officer **R.H. STOPFORD** Telephone No : (02) 6283 2177

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/SG02/00027

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report		Patent Family Member
US	6161139	NONE
US	6023765	NONE
US	5797128	NONE

END OF ANNEX